Table of Contents

BOOK I

Table of Contents (Continued)

SUBJECT **PAGE**

Table of Contents (Continued)

BOOK II

SUBJECT **PAGE**

Table of Contents (Continued)

SUBJECT **PAGE**

List of Figures

BOOK I

List of Figures (Continued)

BOOK II

List of Figures (Continued)

List of Figures (Continued)

List of Tables

BOOK I

List of Tables (Continued)

BOOK II

[Nintendo Licensee Letterhead{
[Insert Date]

Re: Confidentiality Agreement

Dear _____:

_____(the "Company") is a licensee of certain confidential, proprietary, and trade secret information belonging to Nintendo of America Inc. ("Nintendo"). Such information of Nintendo may be provided to you by the company or directly by Nintendo in reliance on your relationship to the Company and your agreement expressed herein. All references in this letter to confidential, proprietary, and trade secret information will be deemed to refer solely to confidential, proprietary, and trade secret information of Nintendo and its affiliated corporations.

Your obligations in connection with confidential, proprietary, and trade secret information of the Company which are reflected in other agreements between you and the Company, remain unchanged, and are in full force and effect.

In consideration of the disclosure of confidential, proprietary, and trade secret information to you, you agree that, except as required by your services to the Company, you will not, at any time during the term of your association with the Company, or at any time thereafter, directly or indirectly use, communicate, disclose, disseminate, discuss, lecture upon, or publish articles concerning such confidential, proprietary, and trade secret information without the prior written consent of the Company.

"Confidential, proprietary, and trade secret information" as used herein means any and all information concerning: (i) copyrights, patents, and/or patent applications owned by Nintendo which are applicable to the Nintendo Entertainment System ("NES"), Game Boy hand held video system ("Game Boy"), Super Nintendo Entertainment System (Super NES), or other hardware, accessory, or software products of Nintendo, (ii) the design, and operation of the NES, Game Boy, Super NES, or other Nintendo products, including without limitation the security system of such products, and (iii) new products, marketing plans, know-how, techniques, and methods relating to the development of software for the NES, Game Boy, Super NES, or other Nintendo hardware, accessory, or software products disclosed to you as a consequence of, or during your association with the Company. Such confidential, proprietary, and trade secret information will not include information which is: (i) a part of the public domain; or (ii) obtained by you from someone otherwise authorized to disclose such information.

All documents, tapes, computer records, notebooks, work papers, notes and memoranda containing confidential, proprietary, and trade secret information, made or compiled by you at any time, or made available to you during the term of your association with the Company, including all copies thereof, will be the property of the Company, and will be held by you in trust and solely for the benefit of the Company, and will be promptly delivered to the Company upon termination of your association with the Company or at any time upon request by the Company.

In the event of any material breach by you of your obligations under this agreement, then the Company will be entitled to such relief, including injunctive relief and damages, including attorney's fees, as may be awarded by a court of competent jurisdiction, in addition to all other relief available to the Company. In the event the Company fails to take action against you for such a breach, Nintendo will have a direct right of action against you, without the necessity of naming the Company.

Preface

TECHNICAL QUESTIONS

If you need technical assistance with your Nintendo Licensee product, our Licensee Support Group Engineers are available between 9:00 a.m. and 6:00 p.m. Pacific Standard Time.

Telephone: 1-206-861-2715

Fax: 1-206-882-3585

Written Inquiries: Nintendo of America Inc.
Engineering Department
Licensee Support Group
4820 150th Ave. N.E.
Redmond, Wa. 98052

CONFIDENTIALITY

Pursuant to the terms of each Nintendo product license and/or confidentiality agreement, Nintendo licensees and developers are required to secure the confidential treatment of information received or derived from Nintendo from all employees, agents, or contractors.

In response to the request of several licensees, we have prepared a supplemental confidentiality agreement which is intended to cover only Nintendo derived information that may be used in your business in addition to confidentiality agreements which you will sign with your employees, agents, and contractors for your own benefit. A sample agreement is included at the end of the Preface for your information.

This supplemental agreement is a suggested format only and is not a required form, as laws in your state or jurisdiction may vary. You may wish to consult with your own legal counsel regarding recommended formats for your state/country. In many cases, your existing confidentiality agreements will protect both Nintendo and you fully. However, we urge each of you to review agreements that you have in place and consider this supplemental agreement, or other supplements, as may be appropriate or necessary to protect the rights of Nintendo.

If you do not presently have a confidentiality agreement in place for your own employees, agents, or contractors, including those who have access to confidential information of Nintendo, we suggest you contact your legal counsel for advice on proper agreements to protect your valuable confidential information and to insure that you are fully in compliance with your Nintendo license/confidentiality agreement.

Please contact our Legal or Licensing Departments at 1-206-882-2040 between 9:00 a.m. and 6:00 p.m. Pacific Standard Time, with any questions you may have concerning this matter.

The obligations set forth in this letter regarding treatment of confidential, proprietary, and trade secret information are continuing obligations which will continue regardless of your continuing association with the company.

Please acknowledge your understanding and acceptance of the foregoing by signing and returning two (2) copies of this letter to the Company for the benefit of the Company and Nintendo.

Yours sincerely,
[Insert Nintendo Licensee/Developer Name]
By:_____

ACKNOWLEDGED AND ACCEPTED
[Insert Contractor or Employee Name]
By:_____
Date:_____

Chapter 1. NOA Licensed Software Approval Process

This chapter describes the process adopted by Nintendo of America Inc. (NOA) which affords interested parties to become Nintendo Authorized Software Developers and/or Nintendo Authorized Software Licensees. The normal process is summarized below.

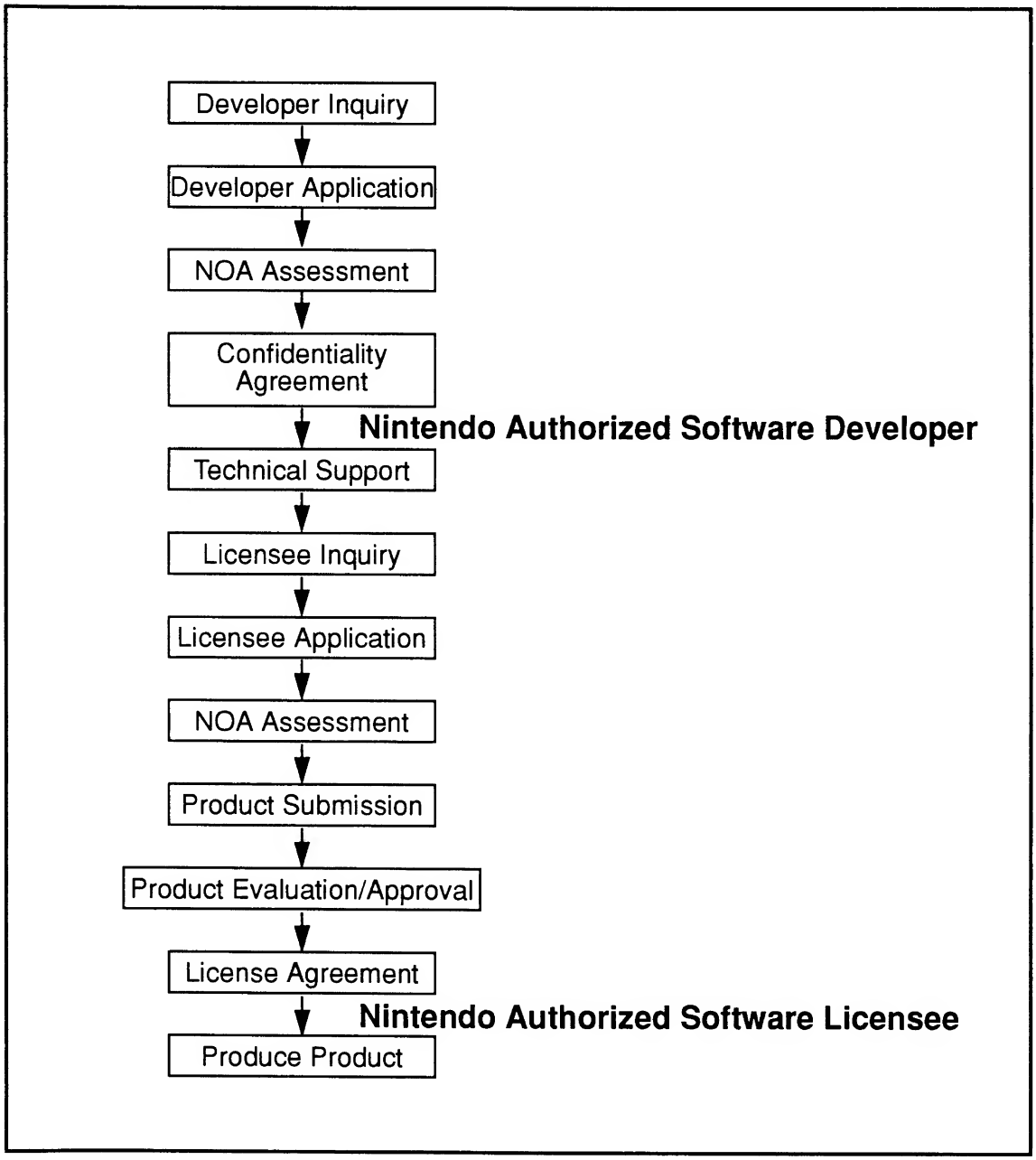


Figure 1-1-1 Software Approval Process

General requirements for the steps listed above are covered in the following paragraphs. Specific questions not answered within this manual should be addressed to NOA's Licensing Department.

1.2 AUTHORIZED SOFTWARE DEVELOPER REQUIREMENTS

Parties interested in becoming a Nintendo Authorized Software Developer may contact the NOA Licensing Department via telephone, FAX, or in writing.

Written Inquiries:	Nintendo of America Inc. Licensing Department 4820-150th Avenue N.E. Redmond, WA 98052 USA
Telephone:	(206) 882-2040
FAX:	(206) 882-3585

In response, the NOA Licensing Department will send a letter which describes specific requirements for becoming a licensed developer. These requirements are described, in general, below.

1.2.1 LETTER OF APPLICATION

A prospective developer's letter of application should include the following items.

1) A detailed description of the submitting individual or company, including a summary of software development or related experience, financial stability, and market leadership. This information should be in the form of a prospectus, business plan, or summary statement.

2) A detailed introduction to key personnel and developers setting forth any technical, managerial, or development experience which may be relevant and identifying any particular software for any system for which they have contributed.

3) A description of any relationships or work undertaken for Nintendo third party licensees.

4) A description of business facilities and equipment.

5) A copy of any confidentiality/non-disclosure agreement which the company's employees/agents are required to sign.

6) A complete listing and at least three samples of software previously developed, especially those which incorporate elements important to successful NOA software or similar entertainment software.

1.2.2 NOA ASSESSMENT

NOA will review the material submitted and make a preliminary determination of whether the prospective developer's qualifications will support designation as an authorized software developer for Nintendo. Because authorized developers are provided with highly proprietary information belonging to Nintendo, and because many of Nintendo's licensees rely on recommendations and referrals to authorized developers, Nintendo exercises a very high level of care in approving only a select number of authorized developers. The Licensing Department will contact the prospective developer with the results of NOA's assessment.

1.2.3 CONFIDENTIALITY AGREEMENT

If the prospective developer's qualifications support designation as a licensed developer for Nintendo, NOA will prepare a formal confidentiality agreement for review by the prospective developer. Once this agreement has been formalized and processed by NOA's Licensing Department, the party described within the agreement becomes a Nintendo Authorized Software Developer for the specified product line(s).

1.2.4 TECHNICAL SUPPORT

All technical documentation which is available for the licensed product line(s) will be forwarded to the licensed developer upon formalization of the confidentiality agreement. In addition, access is afforded to NOA's Engineering Department Licensee Support Group and NOA's Product Development and Analysis Department. These two support groups will assist the licensed developer with any situational requirements or specifications which are subject to special product development.

1.3 AUTHORIZED Software LICENSEE REQUIREMENTS

To license a software product once it has been developed, the interested party must market the product through an existing Nintendo Authorized Software Licensee or become a Nintendo Authorized Software Licensee. NOA would prefer that interested parties contact Nintendo early in the development phase of a product. Therefore, the interested party will already be a licensed developer when they apply to become an software licensee. Exceptions will be made, however, for those parties which have already developed a software product and wish to license it with Nintendo. In such cases, the interested party will be processed and approved as a Nintendo Authorized Software Developer first, then processed as a Nintendo Authorized Software Licensee. In either case, the requirements in the following paragraphs will apply.

Parties interested in becoming a Nintendo Authorized Software Licensee should contact the NOA Licensing Department via telephone, FAX, or in writing. In response, the NOA Licensing Department will send a letter which describes specific requirements for becoming an software licensee. These requirements are described, in general, beginning on the following page.

1.3.1 LETTER OF APPLICATION

A prospective licensee's letter of application should include the following items.

1) A detailed description of the company, including a summary of relevant industry experience, financial resources and stability, and industry leadership or market share. This information should be in the form of a prospectus, business plan, or summary statement.

2) A detailed introduction to key personnel and developers setting forth any technical, managerial, or development experience which may be relevant.

3) A marketing plan for the proposed product(s), including wholesale/ retail price points, targeted distribution channels, advertising commitments, consumer service systems, and merchandising.

4) Any market study information on consumer demand for the proposed product(s) which the company may be relying upon.

5) A written description (in general terms) of the proposed product.

6) A complete listing and at least three samples of software previously developed, especially those which incorporate elements important to successful NOA software or similar entertainment software.

1.3.2 NOA ASSESSMENT

NOA will make a preliminary determination if the:

a) Product would compliment our current line of video game products.

b) Company is capable of the distribution and customer service necessary to support a successful product.

c) Product has any special technical requirements.

NOA's Licensing Department will inform the company of the decision made.

1.3.3 TECHNICAL SUPPORT

If NOA decides to proceed, the prospective licensee will be provided with any technical considerations, suggestions, and any specific technical information required. Technical support will be provided throughout the development of the product, as needed. With respect to those parties previously licensed as developers, this will mean continued support; while those parties contacting Nintendo for the first time will receive a set of technical documentation which is related to the proposed product. A formal confidentiality agreement must be formalized prior to the release of support materials, if one is not already on file.

1.3.4 PRODUCT SUBMISSION AND TESTING

Once the proposed product has been developed and tested by the prospective licensee, it should be submitted in accordance with the applicable software submission requirements, "Super NES Software Submission Requirements" are presented in the following chapter. The samples provided will be tested and results forwarded to the prospective licensee. In cases where failure conditions are detected during testing, NOA will require that the area(s) be corrected and the product be resubmitted for testing.

1.3.5 FORMAL LICENSE AGREEMENT

Once the proposed product is approved, NOA will prepare a formal license agreement for the prospective licensee's review and signature. When formalized, this agreement authorizes the licensee to go into production with the specified licensed product.

Chapter 2. Super NES Software Submission Requirements

All software submissions to Nintendo of America Inc. must be forwarded to the attention of NOA Product Testing Supervisor. Otherwise, the submission's placement into the testing queue may be delayed. To help reduce a submission's turn-around time, it is suggested that licensees assign a primary contact person for each software submission. All communications with NOA concerning a submission's testing status should be forwarded through this individual. The contact person should also be responsible for notifying any other interested parties.

When a submission is not approved, NOA may send a videotaped copy of the programming problem(s) which prevent(s) the submission from being approved. This is intended to assist the licensee in analyzing the cause of the software problem. It is the licensee's responsibility to send a copy of this tape to any developer(s) of the software. NOA strongly encourages that copies be sent to developer(s) of the software as quickly as possible.

2.1 SPECIFICATION SHEET AND CHECK LIST

The appropriate Software Specification sheet and the Software Submission checklist must be filled out completely and must be correct for the particular program version.

2.2 PROGRAM ROMs

One (1) set of the game ROM(s) must be submitted for approval. ROM data submitted must be written on the same size ROM(s) which are intended to be used in production. If a submission ROM is not available in the size to be used, the next size smaller should be used (i.e., a 3M program should be submitted on two 2M ROMs). All ROMs submitted must be of the same manufacturer, size, and part number. A label should be attached to each master ROM which lists game code, ROM version, and ROM number. A copy of the game ROMs submitted should be retained by the licensee for reference, as NOA cannot return originals or copies of submitted ROMs.

2.3 EP-ROMs

Submit only the following EP-ROM types for approval.

4M: TOSHIBA TC574000D, SGS THOMPSON M27C4001, HITACHI HN27C4001, MITSUBISHI M5M27C401K, NEC D27C4001, Texas Instruments TMS27C040-JL, TMS27C040-JL4, TMS27C040-JE, TMS27C040-JE4, ATMEL AT27C040-12C, MACRONIX MX27C4000DC-12

8M: ATMEL AT27C080-10DC, AT27C080-12DC, NEC D27C8001 SGS THOMPSON M27C801-120F1

Note: All ROMs must be 200ns or faster for Normal Speed.

All ROMs must be 120ns or faster for High Speed.

2.4 ROM DATA

In addition to the EP-ROMs, a copy of the ROM data must be submitted in binary format on MS-DOS® 3.5 inch disk(s). The size of the file must be equal to the size of the EP-ROM (i.e., one 4 Meg EP-ROM = one 4 Meg file).

2.5 GAME PLAY VIDEO TAPE/RATING CERTIFICATE

A video tape containing complete game play is required unless the product has been rated by the Entertainment Software Ratings Board (ESRB). If the product has been rated by the ESRB, then a copy of the rating certificate must accompany the submission and no video tape is needed.

2.6 SCREEN TEXT

A printed copy of the complete screen text must be submitted.

2.7 INSTRUCTION MANUAL

Complete game play instructions must be submitted.

NOTE: If any of these items are not satisfied, the program will be rejected and will not be submitted into the approval process until all criteria are met.

2.8 SOFTWARE VERIFICATION

The following verification process will significantly improve the probability of approval of your software.

1. The licensing screen on all submissions should state "LICENSED BY NINTENDO".
2. Confirm the Licensing Screen information is correct.
3. Check the spelling on the Licensing Screen and Title Screen, as well as the spelling and grammar in the screen text.
4. Confirm the use of a ™, circle R (®), or circle C (©) where applicable.
5. Run a "Bypass" Test to assure that, when the game is powered up, the Licensing Screen is visible for at least one second, even if any combination of controller buttons are pressed repeatedly. Also "Power-up" the software repeatedly to assure it does so without programming failures.
6. Game characters should be moved in all possible directions or positions, regardless of whether it is required to play the game properly. For instance, if the game does not require going to a particular area to complete the game, go there anyway to assure there are no programming problems in going to that location.
7. The software should be paused many times during the test, as this often causes programming problems to surface.
8. All testing should be recorded onto a videotape, making it easier to review programming problems.
9. The entire attract mode (demo) should be viewed to assure there are no programming problems.
10. Routines designed to assist the programmer or developer in "debugging" the software should be removed from the game prior to submission. This includes routines to determine hardware type.
11. All references to the Super Famicom, Super Famicom logos, or Super Famicom controllers (with multi-colored buttons) should be removed or revised to represent the Super NES.
12. All games for use with the Super NES Super Scope are required to include a calibration mode.
13. All games are required to have a pause function activated by the "Start" key.

2.8.1 LICENSEE GAME PLAY VIDEO TAPE PASS/FAIL GUIDELINES

1. The licensee game play video tape must be recorded on a VHS tape, Standard Play speed (SP) for clarity.
2. No editing of the tape is allowed.
3. If more than one tape is needed to show the entire piece of software, then when a second tape begins it must show that the player is in the exact same place as when the first tape left off.
4. No codes or "built-up" characters are allowed.
5. All levels or areas must be completed, in succession.
6. Screen text must have correct grammar and spelling.
7. No deviations from NOA Software Standards Policy may be present.
8. The entire ending credits (if any) must be shown.

2.8.2 LICENSING SCREEN INFORMATION PASS/FAIL GUIDELINES

The following Licensing information should be included for all software. This can be displayed on one (1) or two (2) screens.

1. Licensee's software title.
2. Licensee's trademark and copyright notice
   (© 19__ Licensee's name or copyright owner)
3. LICENSED BY NINTENDO

   EXAMPLE:
   Tom's Golf™or®
   © 1992 ABC Corporation
   LICENSED BY NINTENDO

If a blank screen appears for more than two seconds when powered up, Nintendo suggests placing a message or graphic on the screen so that consumers do not think their game is inoperable (e.g., --"Please Wait"--). If a blank screen appears for more than five seconds during game play, a message or graphic should also be placed on the screen.

2.8.3 COMMON PROBLEMS

Some possible problems that may prevent approval of a piece of software include, but are not limited to the following:

1. Lock up of the software.
2. Scrambled blocks or characters appear on the screen.
3. The software won't pause.
4. Your character can get stuck somewhere with no possible way to get out.
5. Scrambled graphics at the edges of the screen when the screen scrolls in any direction.
6. Vowels in the passwords or password entry-system.
7. Colored lines at the top or bottom of the screen.
8. Shifting of the screen in any direction (other than normal scrolling).
9. Inconsistent scoring methods.

10. Flashes on screen.
11. Small flickering lines on the screen.
12. Hit or be hit by an enemy but no damage is incurred.
13. Three (3) or four (4) player game can be started without using a four player adapter.
14. Incorrect Licensing Screen; "Licensed by Nintendo" should appear for all formats.
15. Violation of any Programming Cautions in the product Development Manual.
16. Use of the Nintendo logo or representations of Nintendo products in software without license agreement.
17. The use of the term Super Nintendo or Nintendo when the Super Nintendo Entertainment System or Nintendo Entertainment System is the intended reference, respectively.
18. Character actions are inconsistent (for instance, a character that cannot fly, being able to walk off the edge of a platform and stand in midair).
19. Referring to the Nintendo control pad by an unacceptable term, such as; "joypad", "directional control", etc.
20. Referring to the Nintendo Controller by an unacceptable term, such as; "joystick", etc.
21. Referring to the Nintendo game pak by an unacceptable term, such as; "Game Cassette", etc.
22. Note: If Licensor approval is required, please assure that this has been finalized before the software submission has been made.
23. Display of Super Famicom symbols or controllers in Super NES games.

2.8.4 A NOTE ON OBJECTIONABLE MATERIAL

A copy of the Nintendo "Game Content Guidelines" is included in book 2 of this manual. If you are unsure of whether an item of text or element of a game is within Nintendo Software Standards, you may contact our Product Analysis Department early in the development process and they will go over questionable items over the phone. In cases concerning an extensive amount of text, please send it to the attention of NOA Product Testing Supervisor, using the address listed in the Preface of this manual, with the questionable items highlighted. The material will be evaluated and you will be contacted within a week to ten days.

SOFTWARE SUBMISSION CHECK LIST

MACHINE TYPE ☐ SNS ☐ VUE ☐ DMG

GAME NAME _____

COMPANY _____

GAME CODE SNS__ __ __ __ VUE__ __ __ __ DMG__ __ __ __

VERSION

- ☐ Evaluation
- ☐ Approval Ver. ____
- ☐ Specification Sheet
- ☐ 1Set of ROMs (These must be specific EP-ROM type. See Submission Requirements.)
- ☐ MS-DOS® 3 1/2 Disk(s) (Files must be in binary format. See Submission Requirements for specific information.)
- ☐ 1 copy of Custom DSP IC if applicable (Super NES Submissions Only)
- ☐ 1 Copy of VHS Tapes or ESRB Rating Certificate
- ☐ Screen Text
- ☐ Instruction Manual or Game Play Instructions

REMARKS

_____

_____

_____

_____

NOTE: This check list must be included with the software submission. If any of these items are not satisfied, the program will be promptly returned and will not be submitted into the approval process until all criteria are met.

THIS PAGE INTENTIONALLY
LEFT BLANK

Super NES Software Specification

Game Title	
Product Code	SNS - ___ ___ ___ ___
Accessories	☐None ☐Super Scope ☐Super NES Mouse ☐MultiPlayer 5 ☐Other ( )
Overseas Version	☐No ☐Yes Game Title: _____ Country: _____ Release Date: _____
Company	
Department	
Contact Name	
Address	_____ _____ **Tel:** _____ **Fax No.:** _____
Submission Date	___/___/___ M D Y **Method of Submission:** ☐Mail ☐By Hand

ROM Registration Data

Data Name	Address	Data	Data Name	Address	Data
Maker Code	FFB0H	____H (' ____')	Game Title Registration	FFC0H~ FFD4H	
	FFB1H	____H (' ____')	Map Mode	FFD5H	____H
Game Code	FFB2H	____H (' ____')	Cartridge Type	FFD6H	____H
	FFB3H	____H (' ____')	ROM Size	FFD7H	____H
	FFB4H	____H (' ____')	RAM Size	FFD8H	____H
	FFB5H	____H (' ____')	Destination Code	FFD9H	____H
Fixed Value	FFB6H~ FFBCH	00 H	Fixed Value	FFDAH	33 H
Expansion RAM Size	FFBDH	____H	Mask ROM Ver.	FFDBH	____H
Special Version	FFBEH	____H	Complement Check L	FFDCH	____H
Cartridge Type (Sub-number)	FFBFH	____H	Complement Check H	FFDDH	____H
			Check Sum L	FFDEH	____H
			Check Sum H	FFDFH	____H

* Write equivalent letter in parenthesis "( )".

Game Title Registration

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
FFC0	Game Name																
	Code (ASCII)																
FFD0	Game Name						Data Registration										
	Code (ASCII)																

* Use code 20 (H) to fill space and unused area.

(Continued on reverse side)

ROM Version

Mask ROM	☐0 ☐1 ☐2 ☐3 ☐___
EP-ROM	☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐__ ☐E (Interim)

Memory Configuration

ROM	Size: ____ MBit	High Speed Required? ☐ Yes ☐ No
RAM	☐ No ☐ Yes	Size: Bit Battery Back Up? ☐ No ☐ Yes
External Co-processor	☐None	☐ DSP (DSP___) ☐ Super FX (Expansion RAM ____ Bit) Data back-up: ☐ Yes ☐ No ☐ Super FX2 (Expansion RAM ____ Bit) Data back-up: ☐ Yes ☐ No ☐ SA-1 Internal RAM Data Back-up: ☐ Yes ☐ No ☐ Other:_____

Check Sums

EP-ROM Configuration	___MBits x___ Pcs x 1 Set Manufacturer: _____		Model No.: _____	
ROM 0	_____H	ROM 4	_____H	
ROM 1	_____H	ROM 5	_____H	
ROM 2	_____H	ROM 6	_____H	
ROM 3	_____H	ROM 7	_____H	
Total	___ ___ ___ ___ H			

Affix a label to master ROM which contains product Game Code, ROM Version, and ROM Number. Total check sum must be written even though disk media is used.

File Names

Floppy Disk Configuration	3.5" ☐DSHD ☐HD ___ Pcs x 1 Set				
	File Name	HEX Code		File Name	HEX Code
FILE 0	_____	_____H	FILE 1	_____	_____H
FILE 2	_____	_____H	FILE 3	_____	_____H
FILE 4	_____	_____H	FILE 5	_____	_____H
FILE 6	_____	_____H	FILE 7	_____	_____H

Special Programming

Special Programming?	☐ Yes ( ) ☐ No

Remarks:

_____

_____

_____

Instructions for Super NES Software Specification

1. Game Title, Product Code, Scheduled Release Date, Accessories, Overseas Version, Company, Contact, Address, Telephone No., Fax No., submission date and method.

- Product Code (4 digits) will be determined by Nintendo.
- Scheduled release date should be entered.
- Game Title includes sub-title if any.
- Indicate accessories other than standard controller which can be used.
- If the product has been sold, or is to be sold in another country; write the game title, country, and the scheduled release date in that country.
- Company, Contact, Address, Telephone No., Fax No. must be completed.
- Submission date and method of submission should be entered.

2. ROM Registration Data

- Write the contents registered in the indicated addresses of the master ROM. Refer to "Description of ROM Registration Data Specification" for details. Enter ASCII characters in areas marked with parenthesis "( )".

3. Game Title Registration

- Enter the game title registered in the master ROM using ASCII characters and their ASCII codes. Refer to "Character Code List for Game Title Registration".

4. ROM Version

- Mask ROM Version
  The Mask ROM Version number starts from 0 and increases for each revised version sent for changes after starting production.
- EP-ROM Version
  The EP-ROM Version number starts from 0 and increases for each revised version sent for approval.
- Example

	First	Second	Third	⇒ Change after first production	Fourth	Fifth
Mask ROM Version	0	0	0		1	1
EPROM Version	0	1	2		0	1
Version on Title Label of ROM	0.0	0.1	0.2		1.0	1.1

⇑ First Production

	⇒ Change after second production	Sixth	Seventh	
		2	2	•••••
		0	1	
		2.0	2.1	

⇑ Second Production ⇑ Third Production

5. Memory Configuration

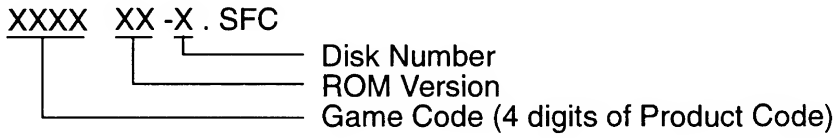
- Enter the memory configuration of the product.
- Enter ROM size and whether or not High Speed Mode (3.58MHz operation) is required.
- RAM
  If RAM is used, enter memory size and indicate whether or not Battery Back-up is used.
- External Co-processor
  If an external co-processor is used (i.e., DSP1, Super FX), select the configuration used.

6. Check Sums

- Enter the check sum of each ROM submitted. To calculate the check sum, add each byte in the ROM data. The lower 2 bytes of the resulting value is the check sum. Enter the check sum for each ROM submitted for the master program and the total of their individual check sums. The total is calculated by simply adding the individual check sums. This method of calculation is different from the check sum on the ROM Registration Specification.

7. File Names

- Write the file name of each disk using the following conventions.



For example,

If the Game Code is AAAE, ROM version is 0.1, and ROM size is 8M; the first disk (Disk 1 of 1) should be named: "AAAE01-0.SFC" (8M file).

If, on the other hand, the Game Code is MW, ROM version is 1.0, and ROM size is 20M;

1st Disk (1 of 3) = "MW_E10-0.SFC" (8M file)
2nd Disk (2 of 3) = "MW_E10-1.SFC" (8M file)
3rd Disk (3 of 3) = "MW_E10-2.SFC" (4M file)

Note that when the Game Code only uses 2 digits, a bar "_" is inserted in the 3rd digit's place and the destination code is inserted in the 4th digit's place.

8. Special Programming

- If special programming is implemented, such as for the purposes of copyright protection, it should be indicated. Also, the contents of the special programming must be explained in writing.

Note: When more than one ROM is required for the game program, all ROMs submitted as a set should be the same part number.

9. Remarks

- If a special configuration of game pak is used, please note the special configuration here. Write the name of the evaluation board which was used for debugging the game. Please write the full name as printed on the board. For example,
  SHVC-4PV5B-10
- If several boards were used for debugging the game, all boards must be listed.

Character Code List for Game Title Registration

	00	10	20	30	40	50	60	70	80	~	F0
0			SP	0	@	P	‘	p			
1			!	1	A	Q	a	q			
2			“	2	B	R	b	r			
3			#	3	C	S	c	s			
4			$	4	D	T	d	t			
5			%	5	E	U	e	u			
6			&	6	F	V	f	v			
7			’	7	G	W	g	w			
8			(	8	H	X	h	x			
9			)	9	I	Y	i	y			
A			*	:	J	Z	j	z			
B			+	;	K	[	k	{			
C			,	<	L	¥	l	|			
D			-	=	M	]	m	}			
E			.	>	N	^	n	~			
F			/	?	O	_	o				

Note 1: Do not use characters in shaded areas.
Note 2: “SP” means space.

Exanple: If ASCII character is A, ASCII code is 41.

ROM Registration Data Specification

1. Insert the game title and Super NES game specification at the specified addresses in the ROM.

2. The ROM Registration Data area is 48 bytes from address 00:FFB0H ~ 00:FFDFH in Super NES Memory.

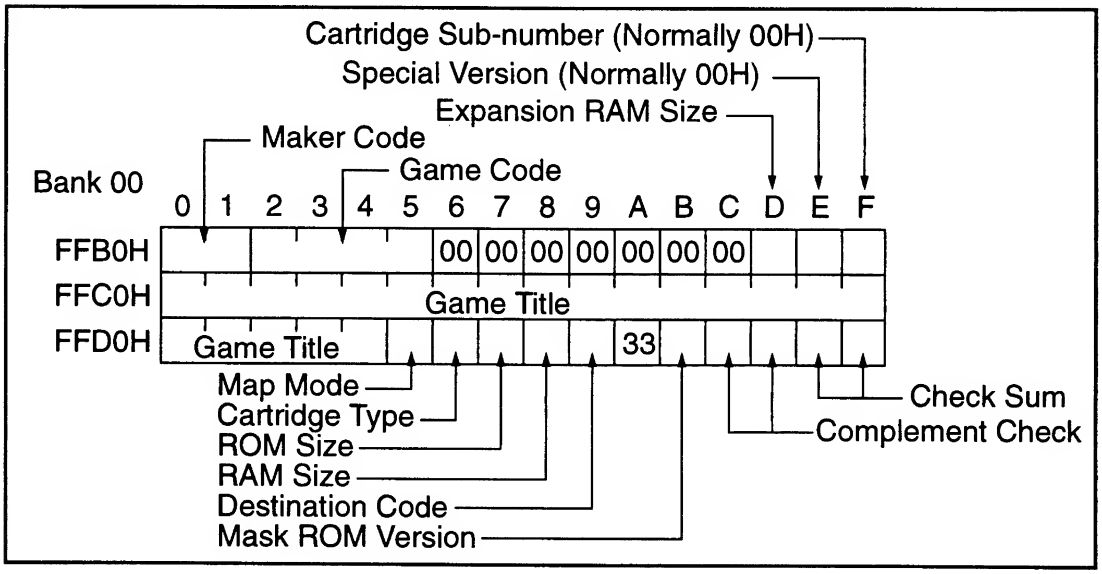
3. The address in ROM for registration data, using Map Mode 20, is 007FB0H ~ 007FDFH.

4. The address in ROM for registration data, using Map Mode 21, is 00FFB0H ~ 00FFDFH.

5. The address in ROM for registration data, using Map Mode 23 (SA-1), is 007FB0H ~ 007FDFH.

6. The address in ROM for registration data, using Map Mode 25, is 40FFB0H ~ 40FFDFH.

7. ROM registration data should be stored using the format below.



8. The following data will be stored in Super NES Memory for *every* Super NES game.

   00:FFB6H ~ 00:FFBCH = 00H
   00:FFDAH = 33H

Description of ROM Registration Data Specification

1. Maker Code (FFB0H, FFB1H)

   Enter the 2-digit ASCII code assigned by Nintendo. Refer to the Nintendo/ Licensee contract, if in doubt. All letters must be in upper case.

   For example;
   If Maker Code is 01, the ASCII code for 0 (30H) is stored at FFB0H and the ASCII code for 1 (31H) is stored at FFB1H.

   If Maker Code is FF, the ASCII code for F (46H) is stored at FFB0H and FFB1H.

2. Game Code (FFB2H ~ FFB5H)

   Enter the 4-digit Game Code assigned by Nintendo in ASCII. All letters must be in upper case.

   For Example;
   If Game Code is "SMWJ", the following ASCII codes will be entered at the indicated addresses.

   53H (S) ⇒ FFB2H
   4DH (M) ⇒ FFB3H
   57H (W) ⇒ FFB4H
   4AH (J) ⇒ FFB5H

   If a game program which was previously assigned a 2-digit Game Code is to be manufactured again, the original 2-digit code will be entered followed by 2 "Space" codes. The ROM submission sheet should be completed in the same manner.

   For example;
   If Game Code is "MW", the following ASCII codes will be entered at the indicated addresses.

   4DH (M) ⇒ FFB2H
   57H (W) ⇒ FFB3H
   20H (space) ⇒ FFB4H
   20H (space) ⇒ FFB5H

3. Fixed Value (FFB6H ~ FFBCH)

   Store fixed value 00H at addresses FFB6H ~ FFBCH.

4. Expansion RAM Size (FFBDH)

Enter the size of the expansion RAM installed in the game pak using the table below. If the size used is not listed below, choose the next larger size which is listed.

For example, enter the size of the RAM used for Super FX co-processor. If no expansion RAM is installed, enter 00H at address FFBDH.

For game paks which use the SA-1, enter 00H at address FFBDH. Enter the size of the RAM used as BW-RAM at address FFD8H.

FFBDH	Size of Expansion RAM
00H	None
01H	16 KBit
03H	64 KBit
05H	256 KBit
06H	512 KBit
07H	1 MBit

5. Special Version (FFBEH)

This is only used under special circumstances, such as for a promotional event. The code 00H should be entered under normal circumstances.

6. Cartridge Type Sub-Number (FFBFH)

This is only assigned when it is necessary to distinguish between games which use the same cartridge type. The code 00H is normally assigned.

7. Game Title (FFC0H ~ FFD4H)

Enter the game title using ASCII code (JIS 8 bit). Refer to "Character Code List for Game Title Registration" for characters which may be used. The code "20H" should be used for a space and for all unused areas. The game title registered should be close to the title under which the game will be marketed, not a temporary name used for development purposes.

8. Map Mode (FFD5H)

This location is used to store the map mode and the speed of operation for the Super NES CPU. Select the appropriate code from the table below.

FFD5H	Map Mode	Super NES CPU Clock
20H	Mode 20	2.68 MHz (normal speed)
21H	Mode 21	2.68 MHz (normal speed)
22H	Reserved-Future Use	----------
23H	Mode 23 (SA-1)	2.68 MHz (normal speed)
25H	Mode 25	2.68 MHz (normal speed)
30H	Mode 20	3.58 MHz (high speed)
31H	Mode 21	3.58 MHz (high speed)
35H	Mode 25	3.58 MHz (high speed)

9. Cartridge Type (FFD6H)

Indicate the game pak (cartridge) configuration. Use one of the tables below, depending upon whether or not a co-processor is used.

Without Co-processor

FFD6H	Game Pak (Cartridge) Configuration
00H	ROM Only
01H	ROM + RAM
02H	ROM + RAM + Battery

With Co-processor

FFD6H		Game Pak (Cartridge) Configuration
Upper	Lower	
0*H	-	Co-processor = DSP
1*H	-	Co-processor = Super FX
2*H	-	Co-processor = OBC1
3*H	-	Co-processor = SA-1
E*H	-	Co-processor = Other
F*H	-	Co-processor = Custom Chip
-	*3H	ROM + Co-processor
-	*4H	ROM + Co-processor + RAM
-	*5H	ROM + Co-processor + RAM + Battery
-	*6H	ROM + Co-processor + Battery

For example;
If a game pak uses the Super FX as its co-processor and contains a 256K Expansion RAM as game pak RAM for battery backup, store 15H at address FFD6H. In this case 05H would be stored at address FFBDH and 00H would be stored at address FFD8H.

If a game pak uses a DSP as its co-processor and no RAM, store 03H at address FFD6H. In this case 00H would be stored at addresses FFBDH and FFD8H.

If a game pak uses the SA-1 as its co-processor with 64K SRAM and battery, store 35H at address FFD6H. In this case, 00H would be stored at address FFBDH and 03H at address FFD8H.

10. ROM Size (FFD7H)

The program ROM size is stored at this address. Select the appropriate code from the table below.

FFD7H	ROM Size
09H	3 ~ 4M Bit
0AH	5 ~ 8M Bit
0BH	9 ~ 16 M Bit
0CH	17 ~ 32M Bit
0DH	33 ~ 64M Bit

11. RAM Size (FFD8H)

The CPU RAM size is stored at this address. Select the appropriate code from the table below. If CPU RAM is not installed in a game pak, store 00H at address FFD8H. If only expansion RAM (game pak RAM) is installed, such as the one used with the Super FX co-processor, 00H is also stored at address FFD8H. The BW-RAM size for an SA-1 game pak should be stored at this address.

FFD8H	RAM Size
00H	No RAM
01H	16K Bit
03H	64K Bit
05H	256K Bit
06H	512K Bit
07H	1M Bit

For example;
If a game pak does not contain a co-processor and uses a 64K RAM for battery backup, store 03H at address FFD8H. In this case 00H is stored at address FFBDH and 02H is stored at address FFD6H.

If a game pak uses the Super FX as its co-processor and contains a 256K Expansion RAM as game pak RAM for battery backup, store 00H at address FFD8H. In this case 05H is stored at address FFBDH and 15H is stored at address FFD6H.

12. Destination Code (FFD9H)

Store the code, from the table below, which best describes where the product will be sold.

FFD9H	Destination (Language)	ROM Recognition Code (Fourth digit of Game Code)
00H	Japan	J
01H	North America (USA and Canada)	E
02H	All of Europe	P
03H	Scandinavia	W
06H	Europe (French only)	F
07H	Dutch	H
08H	Spanish	S
09H	German	D
0AH	Italian	I
0BH	Chinese	C
0DH	Korean	K
0EH	Common	A
0FH	Canada	N
10H	Brazil	B
	Nintendo Gateway System	G
11H	Australia	U
12H	Other Variation	X
13H	Other Variation	Y
14H	Other Variation	Z

13. Fixed Value (FFDAH)

Store fixed value 33H at address FFDAH.

14. Mask ROM Version (FFDBH)

Store the version number of the mask ROM released to the market as a product. The number begins with 0 at production and increases with each revised version.

15. Complement Check (FFDCH, FFDDH)

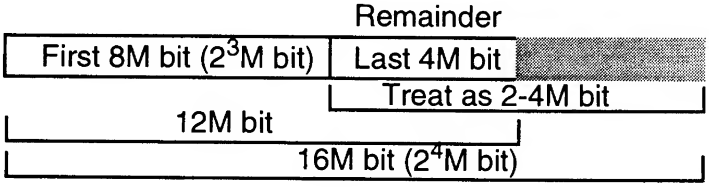
Store the 1's complement of the lower 2 bytes of the program check sum in the order of; FFDCH, lower and FFDDH, upper. Refer to "Check Sum", below, for calculation of the check sum.

(FFDEH, FFDFH) + (FFDCH, FFDDH) = FFFFH
Check Sum Complement Check

16. Check Sum (FFDEH, FFDFH)

First, store 0FFH into the complement check area (FFDCH, FFDDH) and 00H into the check sum area (FFDEH, FFDFH). Then add each byte in the ROM data. If ROM size cannot be expressed evenly in 2^nM bit, such as 10M or 20M bit, add the remainder until a total of 2^nM bit is reached.

For example, If the program contains 12M bit, perform the calculation as if it were 16M bit as shown below.



(Total of first 8M bit) + [(Total of last 4M bit) x2] = Check Sum

For 10M bit, perfrom the calculation as if it were 16M bit.
(Total of first 8M bit) + [(Total of last 2M bit) x4] = Check Sum

For 20M bit, perform the calculation as if it were 32M bit.
(Total of first 16M bit) + [(Total of last 4M bit) x4] = Check Sum

For 24M bit, perform the calculation as if it were 32M bit.
(Total of first 16M bit) + [(Total of last 8M bit) x2] = Check Sum

Next, store the lower 2 bytes of the check sum value into the check sum area (FFDEH, FFDFH). FFDEH will contain the lower byte and FFDFH will contain the upper byte.

Then, store the lower 2 bytes of the complement check in registers FFDCH and FFDDH.

Data Storage on Floppy Disk

1. Use 3.5" DSHD or HD diskettes in MS-DOS IBM format.

2. File data must be in ROM image binary format and not compressed. The maximum data size on a disk is 8M bit. If the program being submitted is larger than 8M bit, the program should be divided and recorded on multiple disks. The last disk must be written to use the full 8M bit.

3. The file name for the disk is determined as follows;

   for example, "AAAJ01-0.SFC".

4. A seal must be affixed to each disk to specify company name, game title, game code, ROM version, date, and disk number.

5. For SA-1 games, don't split data by even and odd addresses.

Super NES Cartridge PCB List

Production PCB List*1

Part Number	Production PCB	ROM	RAM	Other
22536	SHVC-1A0N	1M/2M/4M/8M	None	
22537	SHVC-1A1B	1M/2M/4M/8M	16K	Batt.
22538	SHVC-1A3B	1M/2M/4M/8M	64K	Batt.
22539	SHVC-1A5B	1M/2M/4M/8M	256K	Batt.
22540	SHVC-1B0N	1M/2M/4M/8M	None	DSP1
24468	SHVC-1B5B	1M/2M/4M/8M	256K	DSP1, Batt

Evaluation PCB List*2

Part Number	Evaluation PCB	ROM	RAM	
22427	SHVC-2P3B	1M/2M/4M/8M	None/64K	Battery & 64K SRAM
21945	SHVC-1P0N	1M/2M/4M	None	
24470	SHVC-2Q5B	1M/2M/4M/8M	None/64K/256K	Battery*4,5
25474	SHVC-4PV5B	4M/8M/12M/16M	None/16K/64K/ 256K	Battery*5
33366	SHVC-4PV7B	4M/8M/12M/16M/24M*7	None/512K/1M	Battery & 1M SRAM
28626	SHVC-8PV5B	4M ~ 32M or 4M ~ 64M	None/16K/64K/ 256K	Battery*5
26011	SHVC-2QW5B	4M/8M/12M/16M	None/64K/256K	Battery*4,5
28625	SHVC-1RA3B6S	4M or 8M	64K or 512K*6	Battery & GSU1
28760	SHVC-4QW5B	1M ~ 32M	None/64K/256K	Battery*4,5
22410*3	SHVC-Multi Checker	1M/2M/4M/8M/16M	None/256K/1M	Battery & 256K SRAM
32321	SHVC-8X7B	4M ~ 32M	None/512K/1M	Battery & 1M SRAM

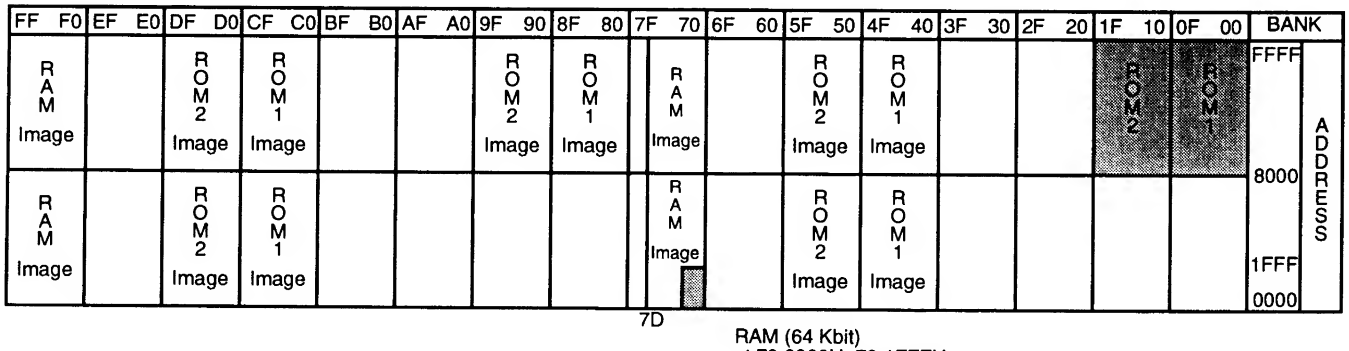
Notes:
1) Mask-ROM should be used on a Production PCB. Production PCBs listed above are bare boards.
2) EP-ROM should be used on an Evaluation PCB. Evaluation PCBs listed above are assemblies.
3) SHVC Multi Checker must only be used with SHVC (Japanese Super NES) in order to evaluate SNS software.
4) DSP1 must be purchased separately.
5) Static RAM(S-RAM) must be purchased separately.
6) The 512K SRAM used with GSU may be configured for battery back-up RAM.
7) 24M requires change of PLD.

21.3 GAME PAK PCB MEMORY MAPPING

The following memory maps are provided for reference. Only the most commonly used memory maps have been included. For information regarding memory maps which are not shown here, contact NOA's Licensee Support Group at (206) 861-2715.

Figure 2-21-1 SHVC-2P3B PCB MEMORY MAP Mode 20 (4M x 2 pcs)

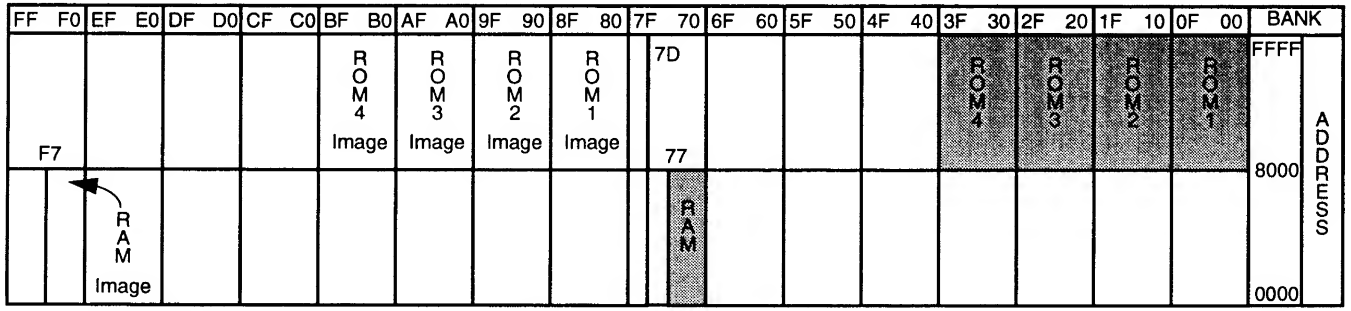
PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM
	20	1/2/4M	1M ~ 8M	64K



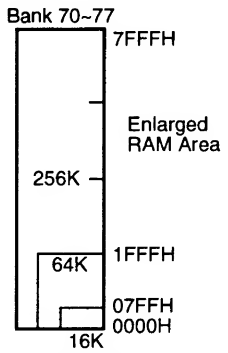
Note 1: Since the area of ROM2 is always started from bank 10, ROM1 and ROM2 will be discontinuous if 1 Mbit or 2 Mbit EPROM is used for ROM1.

Figure 2-21-2 SHVC-4PV5B PCB MEMORY MAP Mode 20 (4M x 4 pcs), Mode 21 (4M x 4 pcs)

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM
	20 or 21	4M/8M*1	4/8/12/16M(8/16/24M*1)	None/16K/64K/256K

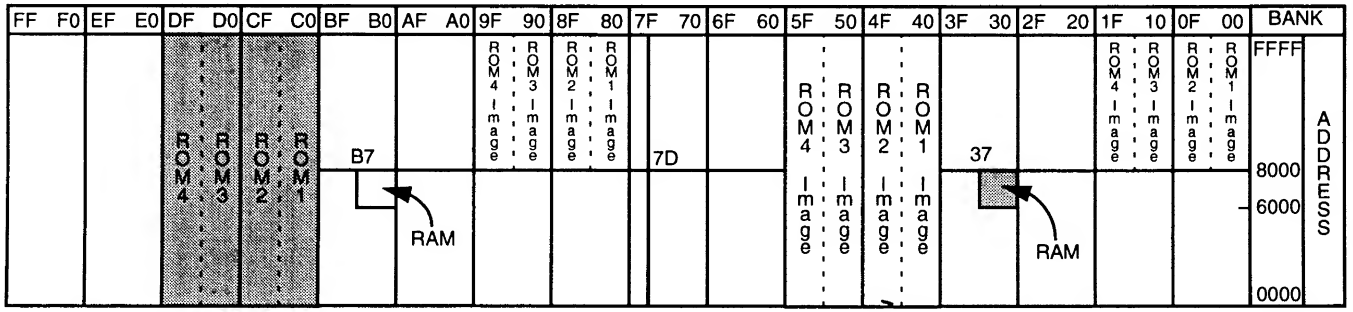


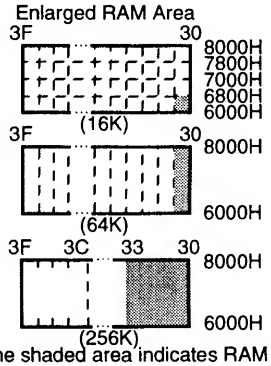
This memory map is used with four 4 Mbit EPROMs.
Note: The RAM image is different from the production PCB.
If you buy PLD, you can use 8 Mbit EPROMs. Maximum ROM size will be 24 Mbits (8 Mbit x 3).



The image of bank 70 is generated in bank 71~7D and F0~FF.

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM
	20 or 21	4M/8M*1	4/8/12/16M(8/16/24M*1)	None/16K/64K/256K

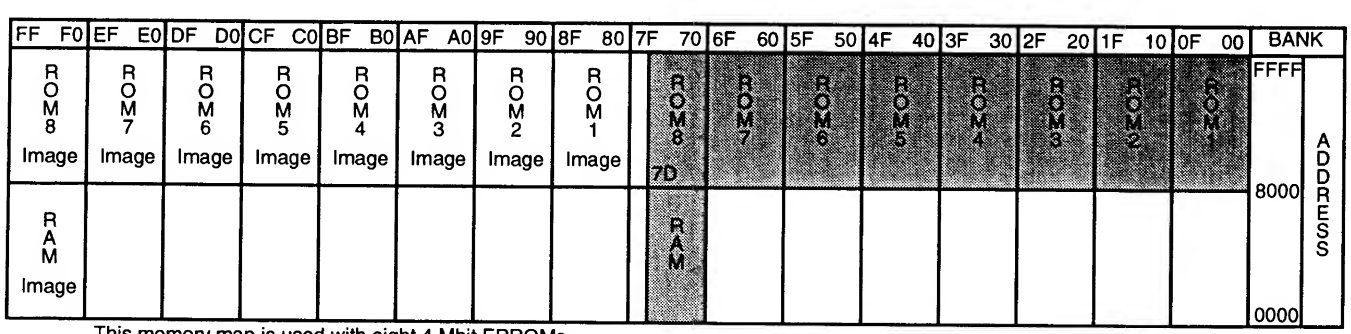




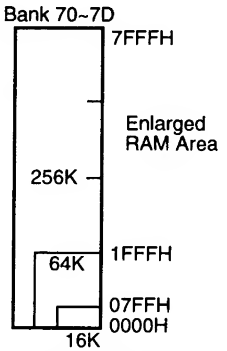
The shaded area indicates RAM area. Dotted area is RAM image.

Figure 2-21-3 SHVC-8PV5B PCB MEMORY MAP — Mode 20 (4M x 8 pcs), Mode 21 (4M x 8 pcs)

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM
	20 or 21	4M/8M	4/8/12/16/20/24/28/32M	None/16K/64K/256K

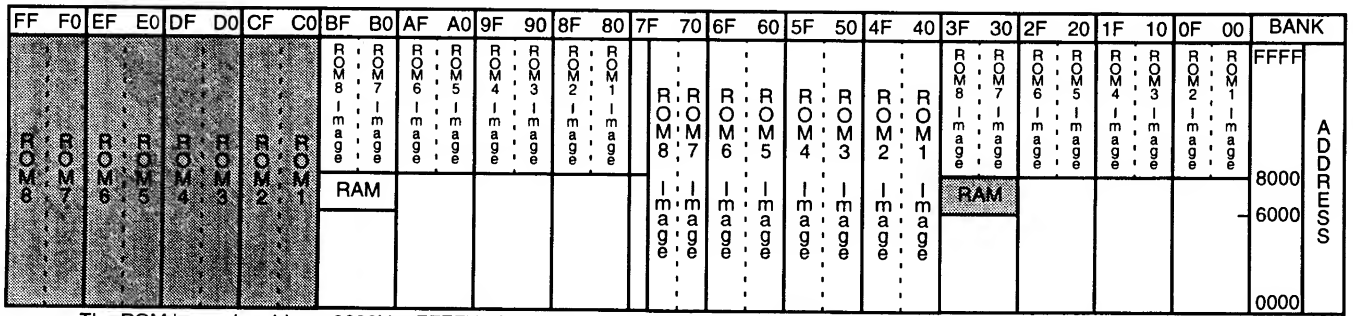


This memory map is used with eight 4 Mbit EPROMs.
Use bank FE and FF for the program ROM area corresponding to bank 7E and 7F.

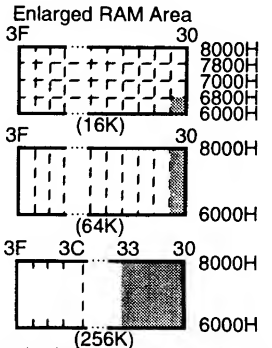


The image of bank 70 is generated in bank 71~7D and F0~FF.

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM
	20 or 21	4M/8M	4/8/12/16/20/24/28/32M	None/16K/64K/256K



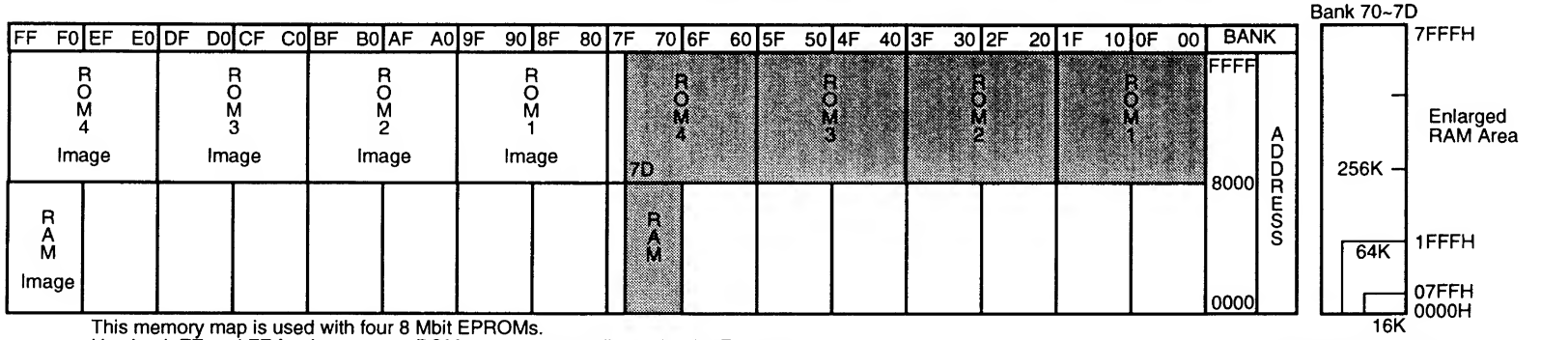
The ROM image in address 8000H ~ FFFFH of bank C0H ~ FFH is generated in bank 00H ~ 3FH and 80H ~ BFH.
This memory map is used with eight 4 Mbit EPROMs.



The shaded area indicates RAM area. Dotted area is RAM image.

Figure 2-21-4 SHVC-8PV5B PCB MEMORY MAP Mode 20 (8M x 4 pcs), Mode 21 (8M x 4 pcs)

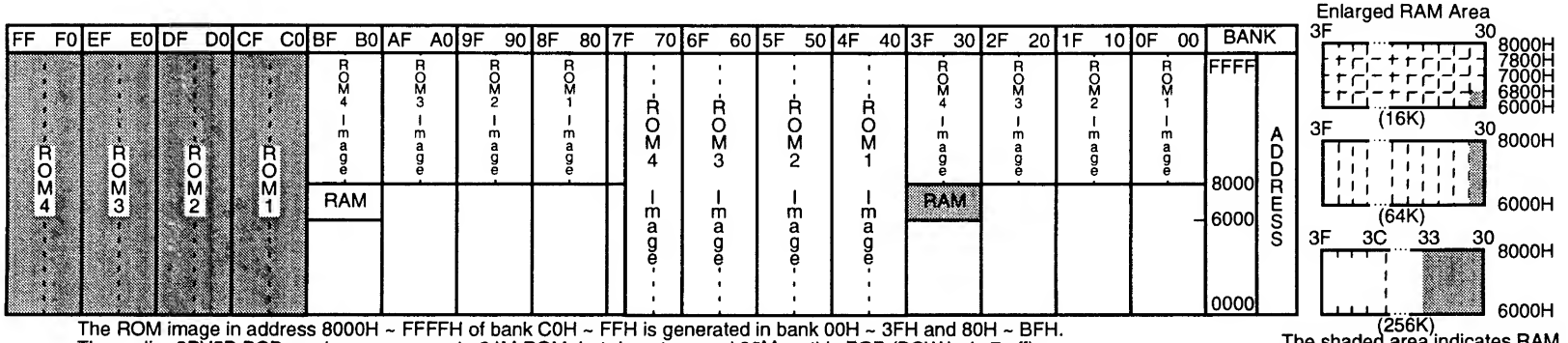
PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM
	20 or 21	4M/8M	4/8/12/16/20/24/28/32M	None/16K/64K/256K



This memory map is used with four 8 Mbit EPROMs.
Use bank FE and FF for the program ROM area corresponding to bank 7E and 7F.
The earlier 8PV5B PCB version can use up to 64M ROM, but do not exceed 32M on this PCB (DSW1 pin 7 off).

The image of bank 70 is generated in bank 71~7D and F0~FF.

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM
	20 or 21	4M/8M	4/8/12/16/20/24/28/32M	None/16K/64K/256K

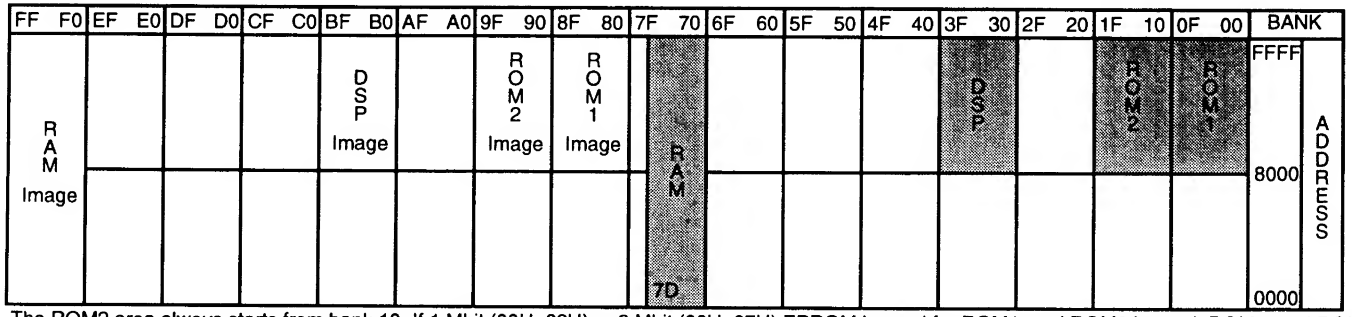


The ROM image in address 8000H ~ FFFFH of bank C0H ~ FFH is generated in bank 00H ~ 3FH and 80H ~ BFH.
The earlier 8PV5B PCB version can use up to 64M ROM, but do not exceed 32M on this PCB (DSW1 pin 7 off).

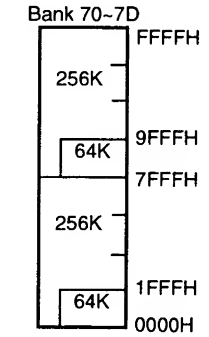
The shaded area indicates RAM area. Dotted area is RAM image.

Figure 2-21-5 SHVC-2Q5B PCB MEMORY MAP Mode 20 (4M x 1 pcs)

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM	Auxillary Device
	20	1M/2M/4M	1/2/4/5/6/8M	None/64K/256K	DSP

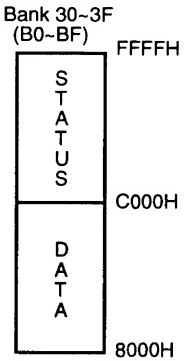


The ROM2 area always starts from bank 10. If 1 Mbit (00H~03H) or 2 Mbit (00H~07H) EPROM is used for ROM1, and ROM2 is used, ROM 1 and ROM2 are not continuous.
The DSP image appears in bank B0~BF, but do not use this area in the high speed mode.
The RAM image appears in bank F0~FF, but do not use this area in the high speed mode.



Enlarged RAM Area

The image of bank 70 is generated in address 0000H~7FFFH and 8000H~FFFFH of bank 71~7D and F0~FF.



Enlarged DSP Area

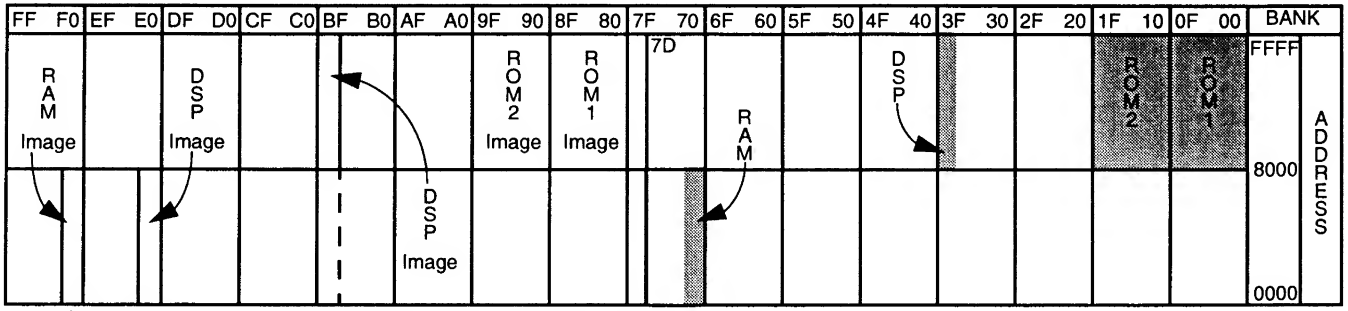
The status image in address C000H~FFFFH of bank 3F is generated in address C000H~FFFH of each bank of DSP area.

The data image in address 8000H~BFFFH of bank 3F is generated in address 8000H~BFFFH of each bank of DSP area.

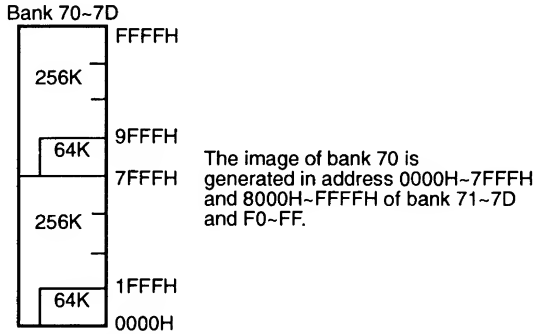
Note: Use C000H/8000H for a port to read from and write to DSP.

Figure 2-21-6 SHVC-2QW5B PCB MEMORY MAP Mode 20 (4M x 2 pcs)

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM	Auxillary Device
	20 or 21	4M/8M	4/8/16M	None/16K/64K/256K	DSP

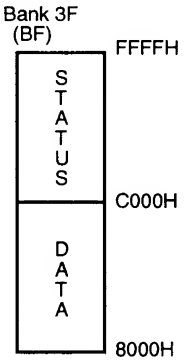


The DSP image appears in bank BF and E0, but do not use this area in the high speed mode.
The RAM image appears in bank F0~FF, but do not use this area in the high speed mode.
Be aware that the DSP area will change, depending upon the ROM size greater or less than 8M, during Mode 20.



Enlarged RAM Area

The image of bank 70 is generated in address 0000H~7FFFH and 8000H~FFFFH of bank 71~7D and F0~FF.



Enlarged DSP Area

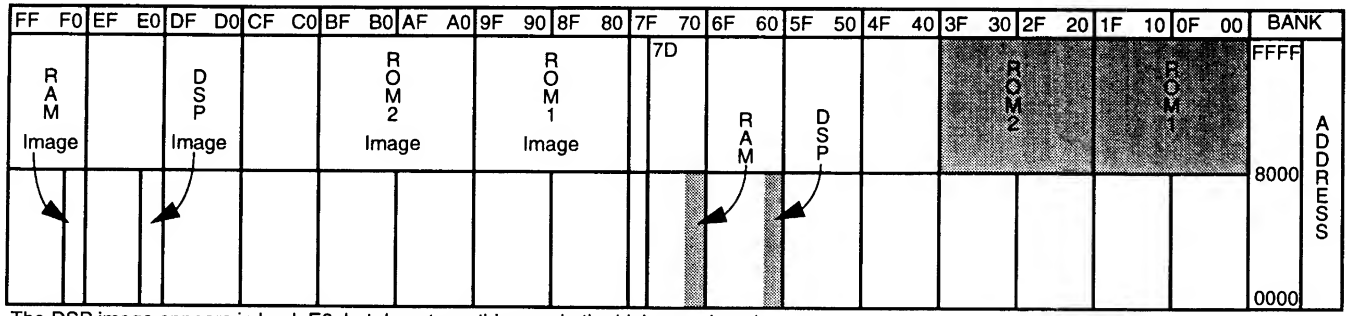
The status image in address C000H~FFFFH of bank 3F is generated in bank BF.

The data image in address 8000H~BFFFH of bank 3F is generated in bank BF.

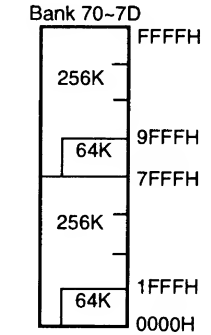
Note: Use C000H/8000H for a port to read from and write to DSP.

Figure 2-21-7 SHVC-2QW5B PCB MEMORY MAP Mode 20 (8M x 2 pcs)

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM	Auxillary Device
	20 or 21	4M/8M	4/8/16M	None/16K/64K/256K	DSP

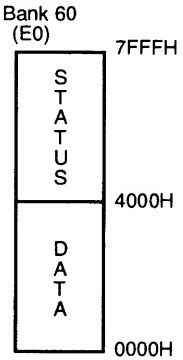


The DSP image appears in bank E0, but do not use this area in the high speed mode.
The RAM image appears in bank F0~FF, but do not use this area in the high speed mode.
Be aware that the DSP area will change, depending upon the ROM size greater or less than 8M, during Mode 20.



Enlarged RAM Area

The image of bank 70 is generated in address 0000H~7FFFH and 8000H~FFFFH of bank 71~7D and F0~FF.



Enlarged DSP Area

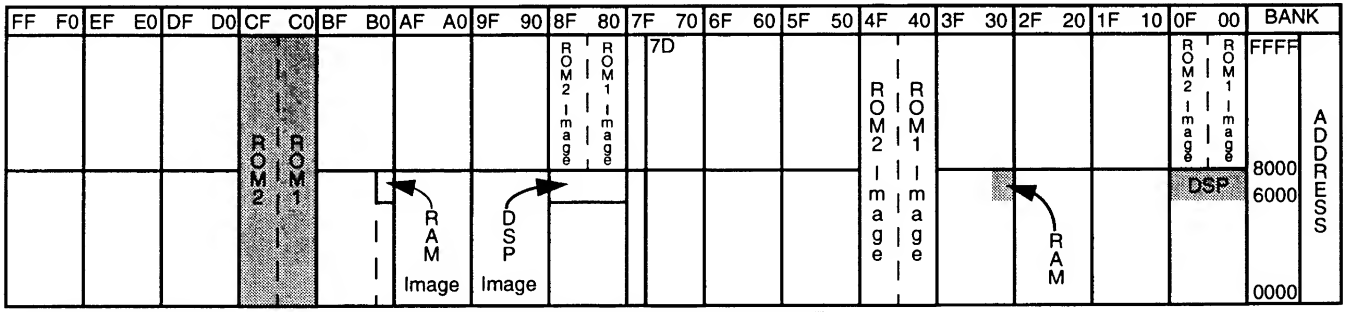
The status image in address 4000H~7FFFH of bank 60 is generated in bank E0.

The data image in address 0000H~3FFFH of bank 60 is generated in bank E0.

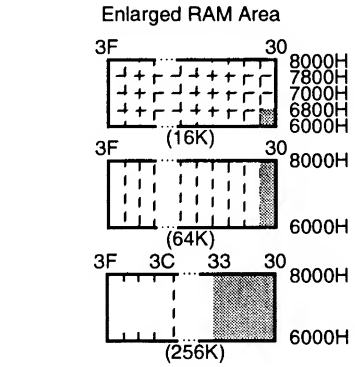
Note: Use 4000H/0000H for a port to read from and write to DSP.

Figure 2-21-8 SHVC-2QW5B PCB MEMORY MAP Mode 21 (4M x 2 pcs)

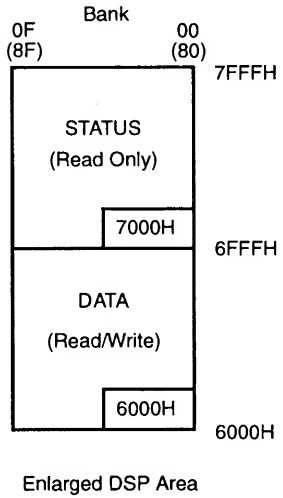
PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM	Auxillary Device
	20 or 21	4M/8M	4/8/16M	None/16K/64K/256K	DSP



The ROM image in address 8000~FFFF of bank C0~CF appears in bank 00~0F and 80~8F.
During Mode 21, the DSP area does not change even if the ROM size changes.



The shaded area is the RAM area, while the dotted area is the RAM area image.



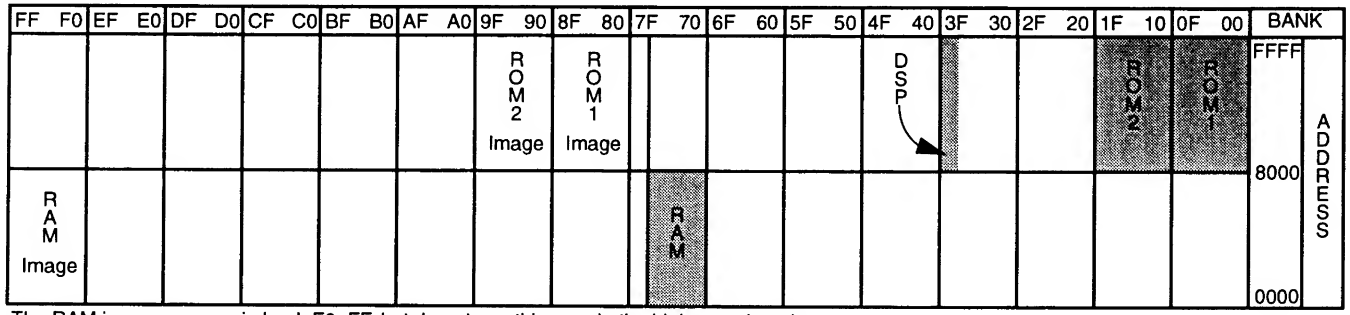
Enlarged DSP Area

The status image in address 7000H~7FFFH of bank 00~0F appears in bank 80~8F.

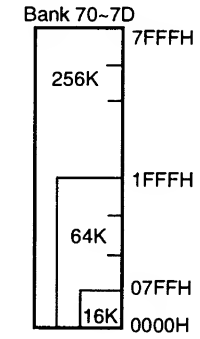
The data image in address 6000H~6FFFH of bank 00~0F appears in bank 80~8F.

Figure 2-21-9 SHVC-4QW5B PCB MEMORY MAP Mode 20 (4M x 2 pcs) ROM Size is 8M or Less

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM	Auxillary Device
	20 or 21	1M/2M/4M/8M	1/2/4/6/8/12/ 16/24/32M	None/16K/64K/256K	DSP

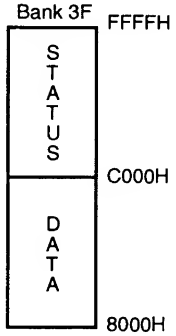


The RAM image appears in bank F0~FF, but do not use this area in the high speed mode.
Ensure that the DSP area is switched to bank 3F when the ROM size is 8M or less during Mode 20 operation. (DSW1 [8] should be On.)



Enlarged RAM Area

The image of bank 70 is appears in address 0000H~7FFFH of bank 71~7D and F0~FF.

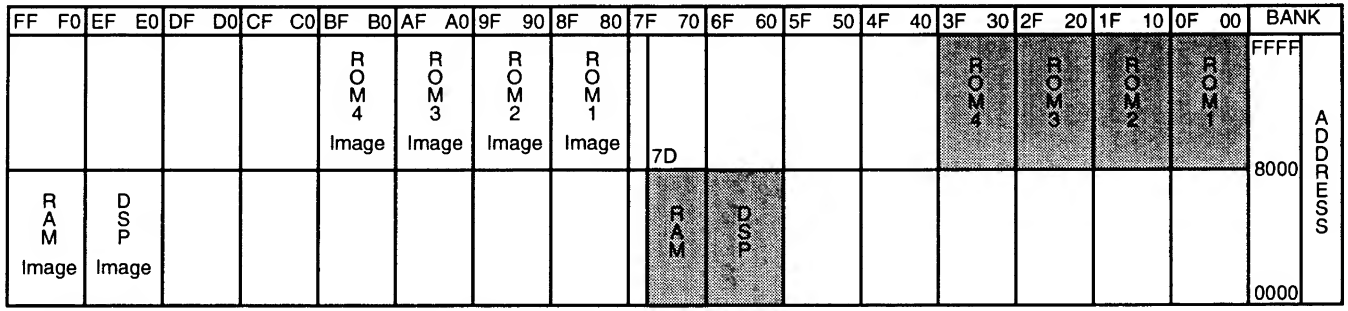


Enlarged DSP Area

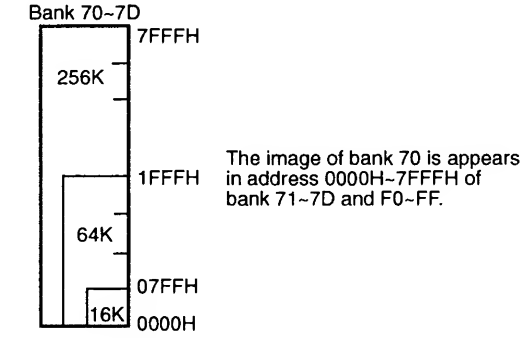
Note: Use C000H/8000H of bank 3F for a port to read from and write to DSP when the ROM size is 8M or less.

Figure 2-21-10 SHVC-4QW5B PCB MEMORY MAP Mode 20 (4M x 4 pcs) ROM Size is 12M or Greater

PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM	Auxillary Device
	20 or 21	1M/2M/4M/8M	1/2/4/6/8/12/ 16/24/32M	None/16K/64K/256K	DSP

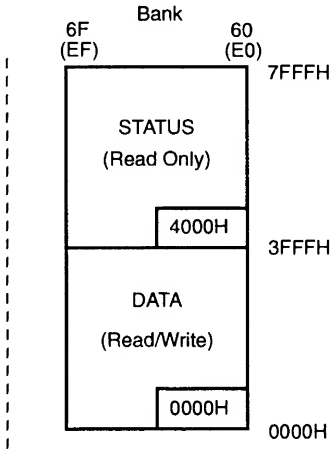


The RAM image appears in bank F0~FF, but do not use this area in the high speed mode.
The DSP image appears in bank E0~EF, but do not use this area in the high speed mode.
Ensure that the DSP area is switched to bank 60 when the ROM size is 12M or greater during Mode 20 operation. (DSW1 [8] should be Off.)



Enlarged RAM Area

The image of bank 70 is appears in address 0000H~7FFFH of bank 71~7D and F0~FF.



Enlarged DSP Area

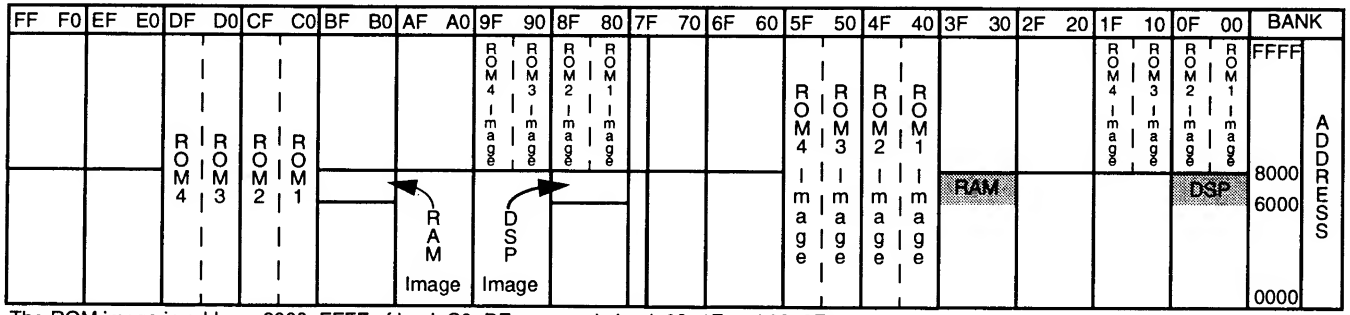
The status image in address 4000H~7FFFH of bank 60~6F appears in bank E0~EF.

The data image in address 0000H~3FFFH of bank 60~6F appears in bank E0~EF.

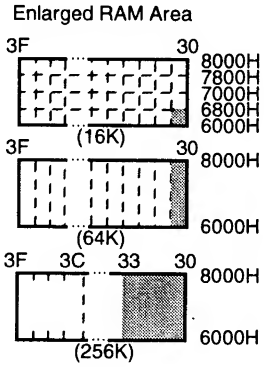
Note: Use 4000H/0000H for a port to read from and write to DSP.

Figure 2-21-11 SHVC-4QW5B PCB MEMORY MAP Mode 21 (4M x 4 pcs)

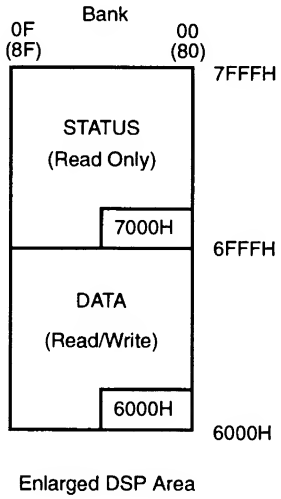
PCB Configuration	Mapping	Usable EPROMs	ROM Size	Usable RAM	Auxillary Device
	20 or 21	1M/2M/4M/8M	1/2/4/6/8/12/ 16/24/32M	None/16K/64K/256K	DSP



The ROM image in address 8000~FFFF of bank C0~DF appears in bank 00~1F and 80~9F.
During Mode 21, the RAM AREA and DSP area do not change even if the ROM size changes.



The shaded area is the RAM area, while the dotted area is the RAM area image.



Enlarged DSP Area

The status image in address 7000H~7FFFH of bank 00~0F is generated in bank 80~8F.

The data image in address 6000H~6FFFH of bank 00~0F is generated in bank 80~8F.

Note: Use 7000H/6000H for a port to read from and write to DSP.

Super NES EPROM Selection Tables

ROM SIZE		1M~8M	4M~16M	4M~16M	1M~32M	4M~32M	8M~64M	1M~8M	4/8/16M	1M~32M	4M~32M	COMMENT FIELD
MODE(S)		20	20/21	20/21	20/21	20/21	20/21/25	20/DSP	20/21/DSP	20/21/DSP	23/SA-1	
STATIC RAM SIZE	None/64K	①	———	———	———	———	———	———	———	———	———	w/back-up
	None/16/64/256K	———	②	———	⑧	④	⑩	⑤	⑥	⑦	———	
	None/512K/1M	———	———	③	———	———	———	———	———	———	⑨	

PCB ASSY		EPROM USED*	REMARKS
①	SHVC-2P3B ASSY	27C1001/27C2001/27C4001	64K SRAM Installed
	Cartridge Evaluation Kit (SHVC-2P3B)	Same as above	25 Units per Kit
②	SHVC-4PV5B ASSY	27C4001/27C8001	Up to 24M by changing PLD
	Cartridge Evaluation Kit (SHVC-4PV5B)	Same as above	25 Units per Kit
③	SHVC-4PV7B ASSY	27C4001/27C8001	Up to 24M by changing PLD 1M SRAM Installed
④	SHVC-8PV5B ASSY	27C4001/27C8001	
⑤	SHVC-2Q5B ASSY	27C1001/27C2001/27C4001	
⑥	SHVC-2QW5B ASSY	27C4001/27C8001	
⑦	SHVC-4QW5B ASSY	27C1001/27C2001/27C4001/27C8001	
⑧	SHVC Multi-Checker 2021	27C1001/27C2001/27C4001/27C8001	256K SRAM Installed
⑨	SHVC-8X7B.ASSY	27C4001	1M SRAM Installed
⑩	SHVC-8PV5B.ASSY-64M	27C8001	

*Note: Use EPROM listed above or one with the same pin locations.

SHVC Cartridge List (20 Map, Production Type)

ROM size		2M	4M	8M	10M	12M	16M	20M	24M	32M	specification
SRAM size	No SRAM	O	O	O	O	O	O	O	O	O	
	16K	O	O	O	O	O	O	△	△	△	
	64K	O	O	O	O	O	O	O	O	O	
	256K	O	O	O	O	O	O	△	△	△	
	512K	O	O	O	△	△	O	△	△	△	
	1M	O	O	O	△	△	O	△	△	△	

[for DSP (77C25)]

ROM size		2M	4M	8M	10M	12M	16M	20M	24M	32M	specification
SRAM size	No SRAM	O	O	O	△	△	△	△	△	△	
	16K	△	△	△	△	△	△	△	△	△	
	64K	O	O	O	O	O	O	△	△	△	
	256K	O	O	O	△	△	△	△	△	△	

O : Now available.
O : In development. Please submit "Price Quote Request for Super NES Cartridge" to the Licensing Department of NOA.
△ : No plan for development at this time. If necessary, please submit "Price Quote Request for Super NES Cartridge" to the Licensing Department of NOA five months prior to the release date.

Note: Back-up RAM sizes of 512K bit and 1M bit are under development. If required, contact NOA Licensing Department.

SHVC Cartridge List (21 Map, Production Type)

ROM size		2M	4M	8M	10M	12M	16M	20M	24M	32M	specification
SRAM size	No SRAM	○	○	○	○	○	○	○	○	○	
	16K	○	○	○	△	○	○	△	△	△	
	64K	○	○	○	○	○	○	○	○	○	
	256K	○	○	○	○	○	○	△	△	△	
	512K	△	△	△	△	△	△	△	△	△	
	1M	△	△	△	△	△	△	△	△	△	

[for DSP (77C25)]

ROM size		2M	4M	8M	10M	12M	16M	20M	24M	32M	specification
SRAM Size	No SRAM	○	○	○	△	△	○	△	○	○	
	16K	○	○	○	○	○	○	△	△	△	
	64K	△	△	△	○	○	○	△	△	△	
	256K	△	△	△	△	△	△	△	△	△	

○ : Now available.
○ :In development. Please submit "Price Quote Request for Super NES Cartridge" to the Licensing Department of NOA.
△ : No plan for development at this time. If necessary, please submit "Price Quote Request for Super NES Cartridge" to the Licensing Department of NOA five months prior to the release date.

Note: Back-up RAM sizes of 512K bit and 1M bit are under development. If required, contact NOA Licensing Department.

THIS PAGE INTENTIONALLY
LEFT BLANK

Price Quote Request for Super NES Cartridge

Please send this form to Nintendo of America Inc. Attn.: Juana Tingdale, Licensing Department by Fax at (206) 861-2173.

Date(M/D/Y)	/ /	Licensee	
Release Date(M/D/Y)	/ /	Game Title	
Quantity		Contact	
		Telephone No.	

Specification

<Map Mode>	20 map / 21 map / to be determined (please circle)
<ROM size>	_____ M Bit
<RAM Specification>	_____ Bit / without RAM
<Backup>	Yes / No
<Co-processor>	DSP1 / μPD77C25 (original program) / other DSP ( ) OBC1 Super FX

(please circle)

Others: Please specify if you are inquiring other than standard specification.

☐ 任天堂記入欄 FOR NCL USE ONLY				業務部
⑨ 業務課受領者	業務課受領者	技術課担当者	部 長	
⑩ 技術課担当者				
⑪ 設計担当部署				
<備 考 欄>				

設計担当部

<コメント>	⑫ 予想開発期間 平成 年 月 日 頃				
	受領署	設計担当	検印	検印	部長
※ EPROM基板の開発予定等・・					

Chapter 1. Introduction

The following is a brief discussion of basic concepts used to display game characters on the home television set. Even if you have developed software for the Nintendo Entertainment System (NES), please review this information.

1.1 PICTURE IMAGE GENERATION

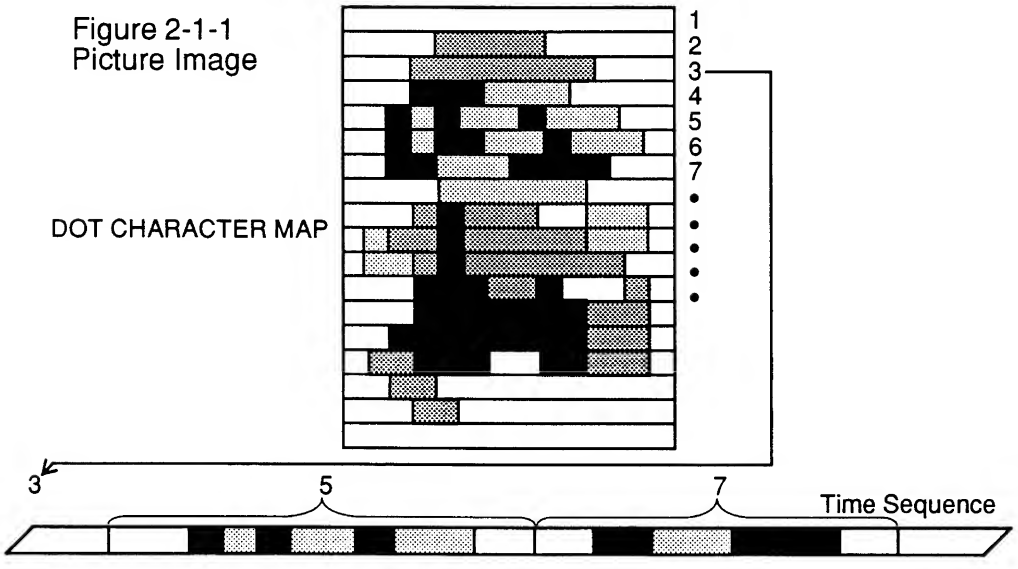
The picture on a color television set consists of 525 horizontal lines with each line having color information. The broadcasting station breaks the picture into lines as shown in the figure below.

The odd numbered lines are converted to electronic signals from the top to the bottom of the screen. The remaining even numbered lines are converted from the top to the bottom in the same way.

This method, in which a trace is generated and displayed for every other line, is called the 'INTERLACE" method. The electronic signal which has been transmitted is converted to a light signal and will create traces on the television screen in the same order generated.

The act of tracing light on the screen is called "scanning". The period while scanning the odd numbered lines is called the "1st field". The period while scanning the even numbered lines is called the "2nd field." A scan period on the screen is called "one frame". During the period of one frame, the first and second fields are displayed in sequence. Because 1/60 of a second is required to produce one field, 1/30 of a second is required to produce one frame. Therefore, a certain point on the screen is radiated only every 1/30th of a second. Due to the afterimage seen by the human eye and the luminescence of the CRT, the picture does not normally appear to flicker.

Figure 2-1-1
Picture Image



(NCL PG 2)

1.2 SUPER NES DISPLAY

The picture display on the Super Nintendo Entertainment System (Super NES) has two modes. One is an interlace mode, based on the television system. The other is a non-interlace mode, in which one frame takes 1/60th of a second. In the non-interlace mode the same position is scanned every field. Each frame consists of only 262 lines, half that of the interlace mode. There appears to be no flickering compared to the interlace mode, since each point on the screen is radiated every 1/60th of a second.

1.3 BLANKING

The screen is scanned from left to the right and from top to bottom (see Figure 1-1-2). After scanning the screen from left to right, horizontal blanking occurs to prevent the electron beam from being seen as it returns to the left side of the screen. When the beam reaches the bottom right hand side of the screen, vertical blanking occurs to allow the beam to reposition at the top left of the screen without being seen. The NES and the Super NES use this blanking efficiently to display the various movements of characters.

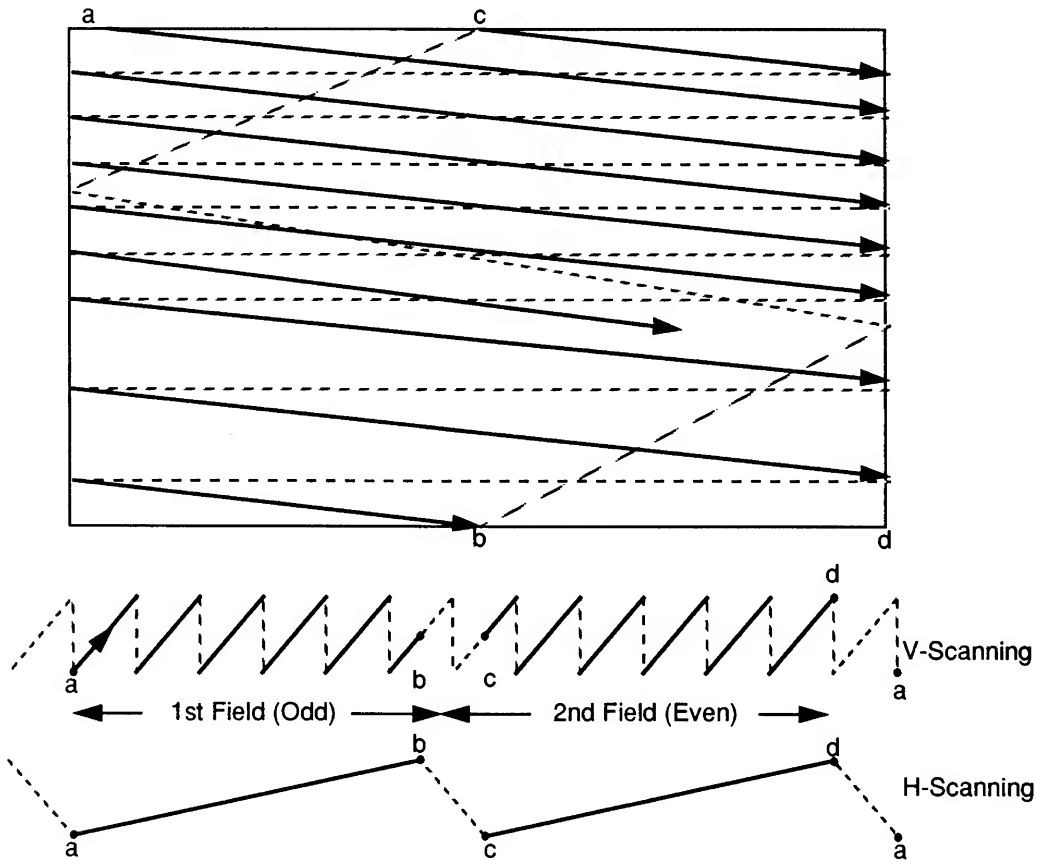


Figure 2-1-2 - Scanning Pattern for Interlace

(NCL PG 3)

Chapter 2. Object (OBJ)

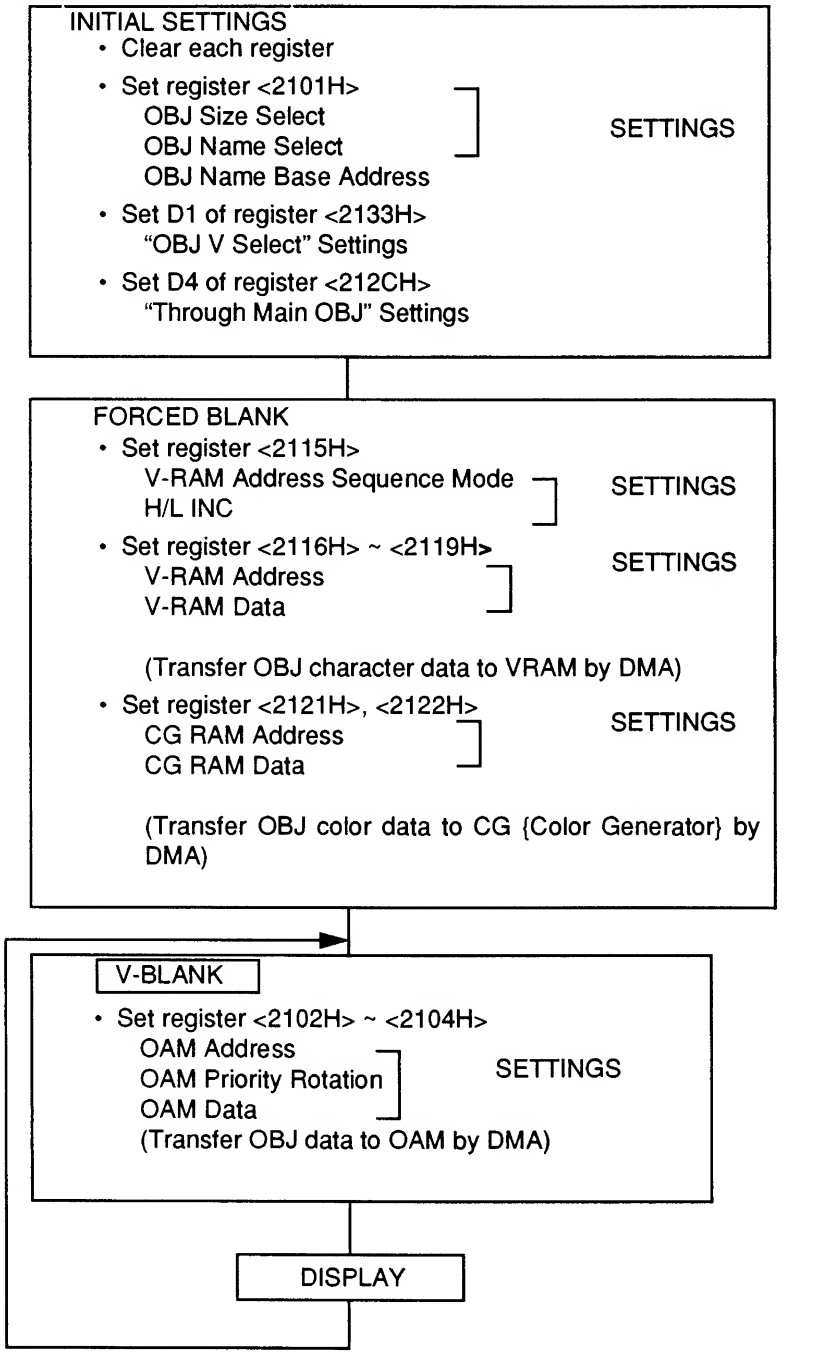
2.1 OUTLINE

This function can display an object in a certain position on the screen. The characters, such as the UFO or the missile of a space game, look like they are moving. If the character's picture is replaced at the same time the point is moved, animation effects can occur (such as "Mario" character looking like he is walking).

2.2 FUNCTION

The maximum number of OBJ's that can be displayed on the screen is 128 and there are four sizes. Two sizes can be selected in one frame and one size can be selected for each OBJ. There are 8 color pallets for OBJ's and one pallet can be selected for each OBJ. One color pallet has 16 color codes out of 32,768 colors. Therefore, each OBJ in the picture is drawn by 16 colors. Each of the 128 objects that may be displayed on the screen at one time has its own priority order, which will decide the display priority if 2 or more OBJ's are overlapped. In addition, there is the Flip function of "up-down," and "left-right," "BG Priority Order" and the "Priority Order" shifting function.

2.3 SETTING EXAMPLE



CAUTION: It is prohibited to write "100H" to the "OAM H-position (9-bit)
(Refer to page A-4)

Chapter 3. Background (BG)

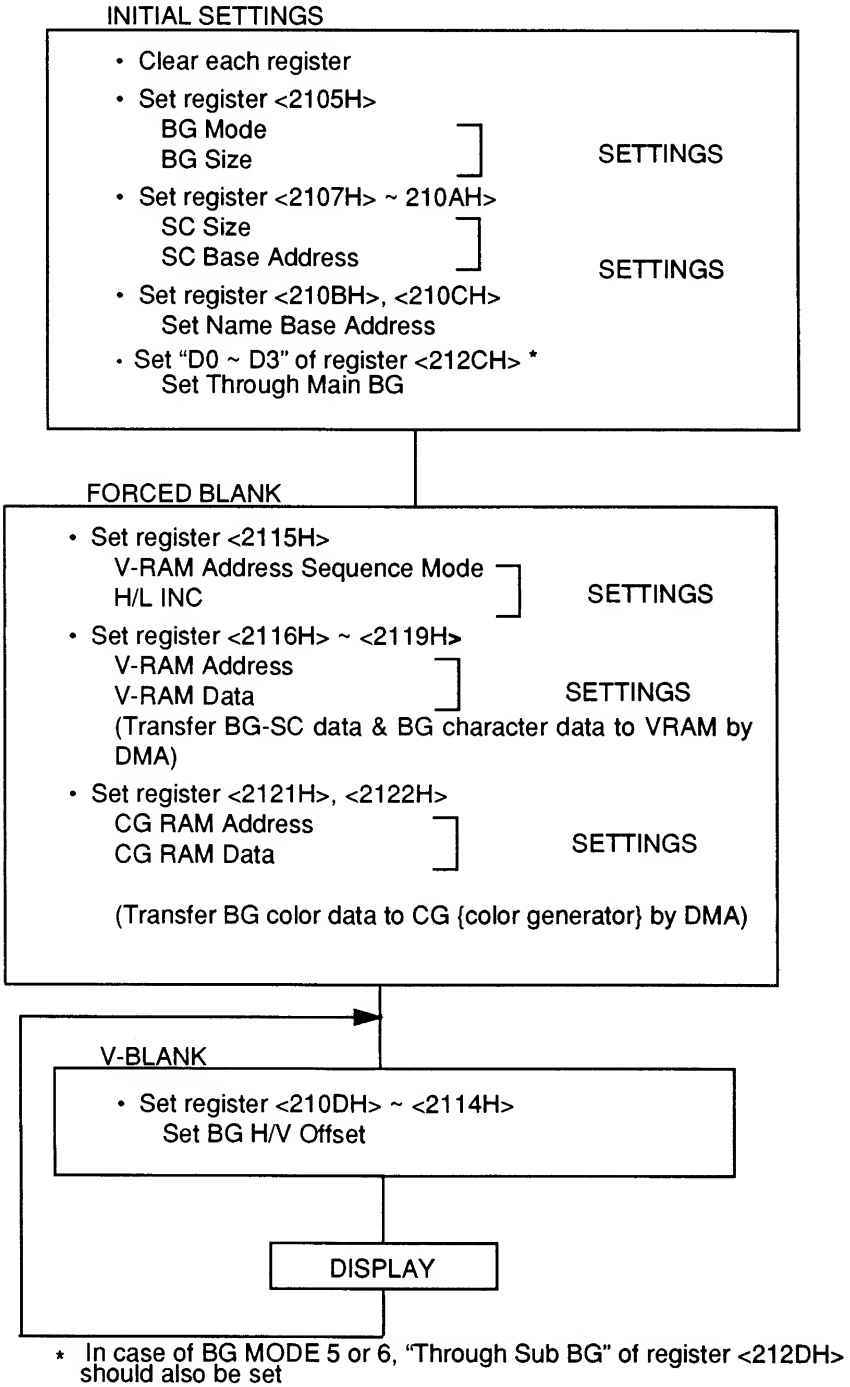
3.1 OUTLINE

The background for OBJ, such as Mario, can be displayed on the screen and scrolled up, down, left or right. This helps the game effect.

3.2 FUNCTION

There are 8 kinds of BG mode. In BG mode 0 thru 6, there is a difference depending on the combination of numbers of screens, the numbers of the cell color, the resolution and the offset function. There are 4 screens provided and the number of the cell colors are 4 to 256. There are 3 kinds of the resolution selected from 256-dot x 224-dot, 512-dot x 224-dot, or 512-dot x 448-dot. The character size can be set "8-dot x 8-dot" or "16-dot x 16-dot" on each screen.
The offset value (scroll coordinate) can be set on each BG screen and the offset value can be changed every horizontal character unit, depending on the mode, so that the vertical partial scroll can be made. Eight pallets can be used per character, and H-Flip or V-Flip is available per character. Also, the priority order of BG and OBJ can be changed per character. (Refer to page A-19)
Mode-7 is a screen, which can rotate, enlarge or reduce. There are other functions for BG, such as mosaic, window, fixed color addition/subtraction, screen addition/subtraction, and H-Pseudo 512.

3.3 SETTING EXAMPLE



* In case of BG MODE 5 or 6, "Through Sub BG" of register <212DH> should also be set

(NCL PG 7)

Chapter 4. Mosaic

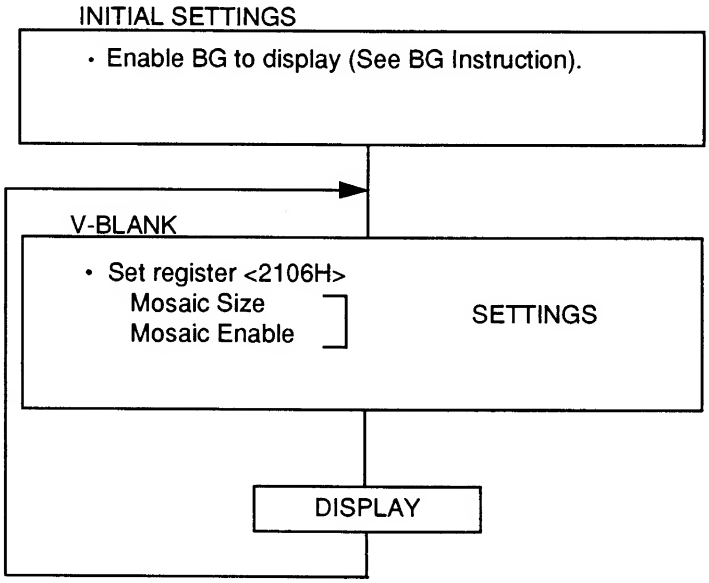
4.1 OUTLINE

The purpose of this function is to change BG screen to mosaic design and shade off a picture (refer to page A-7).

4.2 FUNCTION

A picture element of mosaic design can be changed to 15 sizes and a mosaic design can be selected for BG screen.

4.3 SETTING EXAMPLE



(NCL PG 8)

Chapter 5. Rotation/Enlargement/Reduction

5.1 OUTLINE

In the BG Mode-7 function, more animation effects, are available to the screen through rotation, enlargement or reduction, and a scroll function.

5.2 FUNCTION

5.2.1 TYPE I

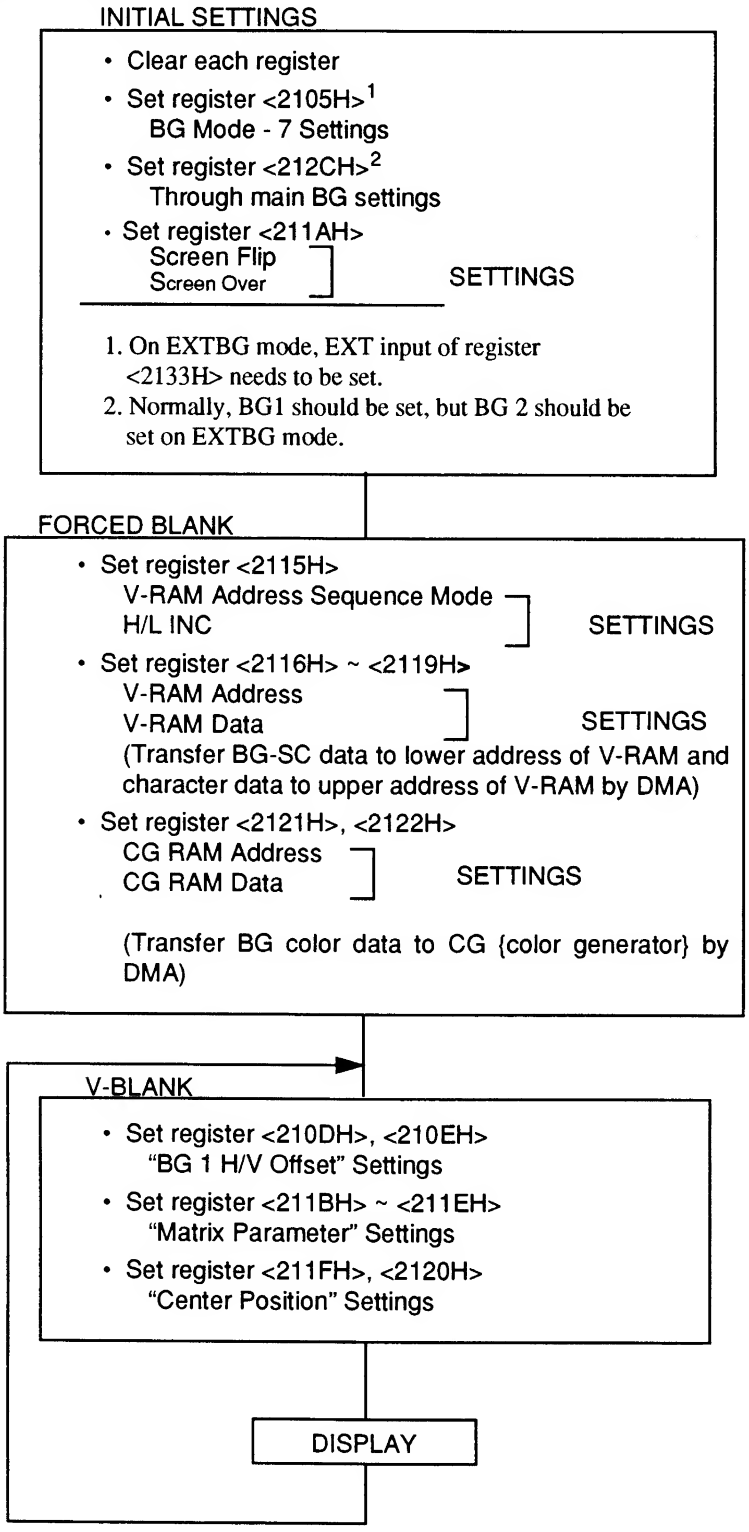
There are 256 character numbers (8-dot x 8-dot size). Each dot can be one of the 256 colors, from a selection of 32,768 colors. In EXTBG mode, each dot can be one of 128 colors from a selection of 32,768 colors and each dot can have priority order. In this function, it is possible to scroll up, down, to the left or right. The center coordinate of rotation, enlargement and reduction can be set at a point either outside or inside of the display area. The rotation angle and vertical or horizontal magnification values are changeable. Also, horizontal flip and vertical flip on the display area are possible. In case the display area goes beyond the screen area, one of three choices can be selected in order to display the excess portion:

1) the back drop color
2) a single character (CHR# 0)
3) repetition (wrap) of the screen area

5.2.2 EXTBG MODE(TYPE II)

EXTBG mode is originally provided as a function for the purpose of the LSI BG expand. For the Super NES, this function is used for rotation, enlargement and reduction in 128 colors with priority order..

5.3 SETTING EXAMPLE



(NCL PG 10)

Chapter 6. Window (Window Mask)

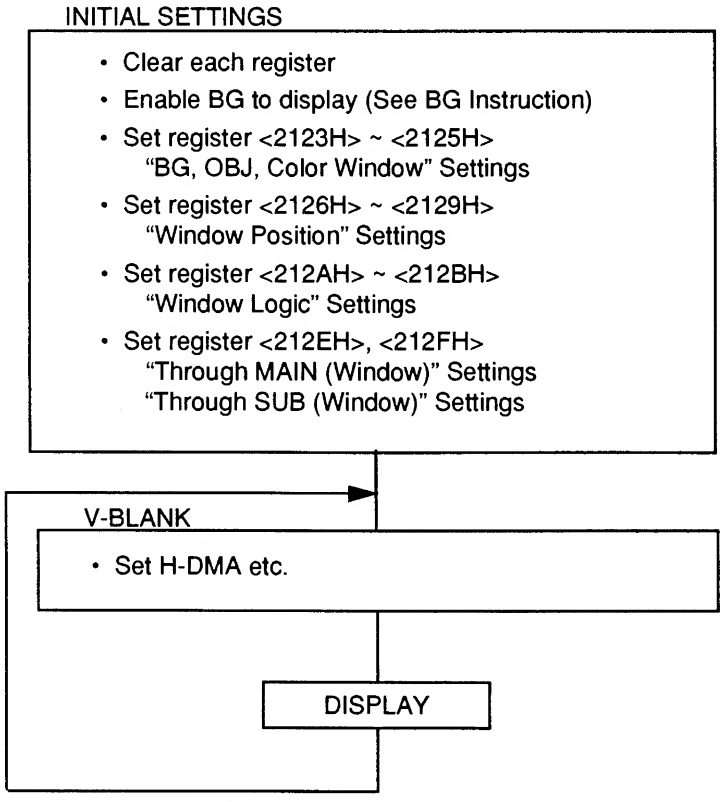
6.1 OUTLINE

This function limits the display area on the TV screen for BG and OBJ. This window can be set on the TV screen. BG and OBJ can be displayed inside or outside of this area.

6.2 FUNCTION

There are 2 windows. Each window can affect either the BG screen or OBJ and can be either internal or external masked. Four types of window mask logic (OR, AND, XOR and NXOR) can be selected for each BG and OBJ, using 2 kinds of windows simultaneously (refer to "Mask Logic Settings for Window 1 & 2" under "PPU Registers"). If this function is combined with the function of H-DMA, various shapes of the window will be formed, such as; a round shape, heart shape, or star shape. It is also possible to use this function combined with the screen addition/ subtraction and fixed color addition functions.

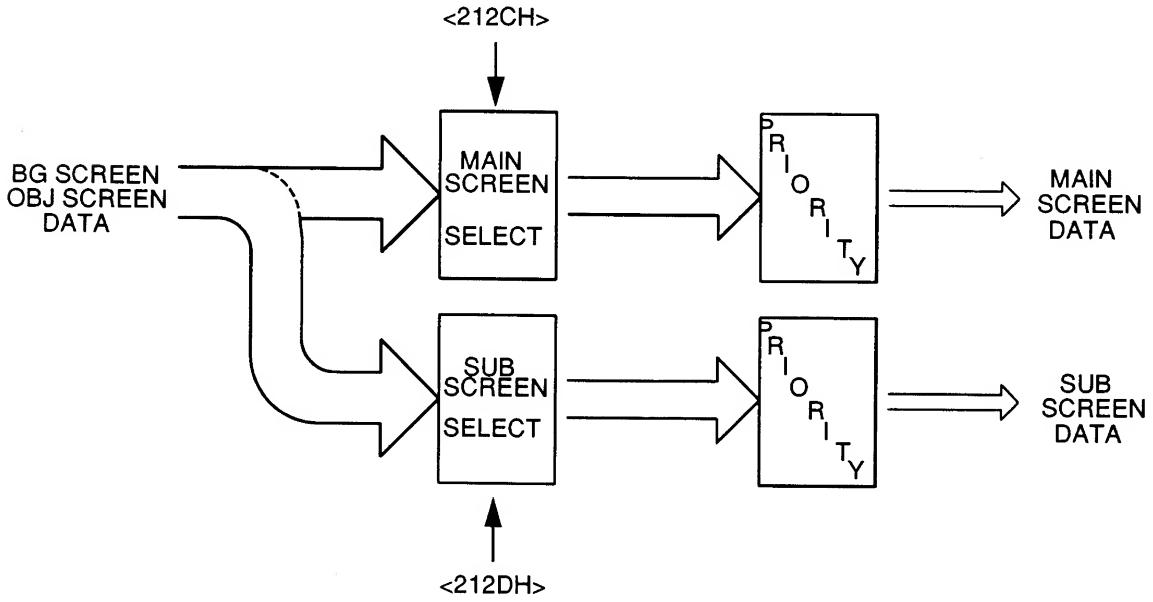
6.3 SETTING EXAMPLE



(NCL PG 11)

Chapter 7. Main/Sub Screen

When displaying several BG and OBJ screens, the picture to be displayed in the overlapped portion is decided by two paths. One of them is called the main screen and the other is called the sub screen. The screen to be used for the main and sub screens can be selected by registers <212CH> and <212DH>. Furthermore, the data for the main and sub screens to be displayed is made according to the priority order. Unless the addition/subtraction screen is done as follows, the "Main SW" of the "Color Window" in register <2130H> is normally on, and the "Sub SW" is normally off so that only the main screen is displayed (see page A-23).



7.1 SCREEN ADDITION/SUBTRACTION

7.1.1 OUTLINE

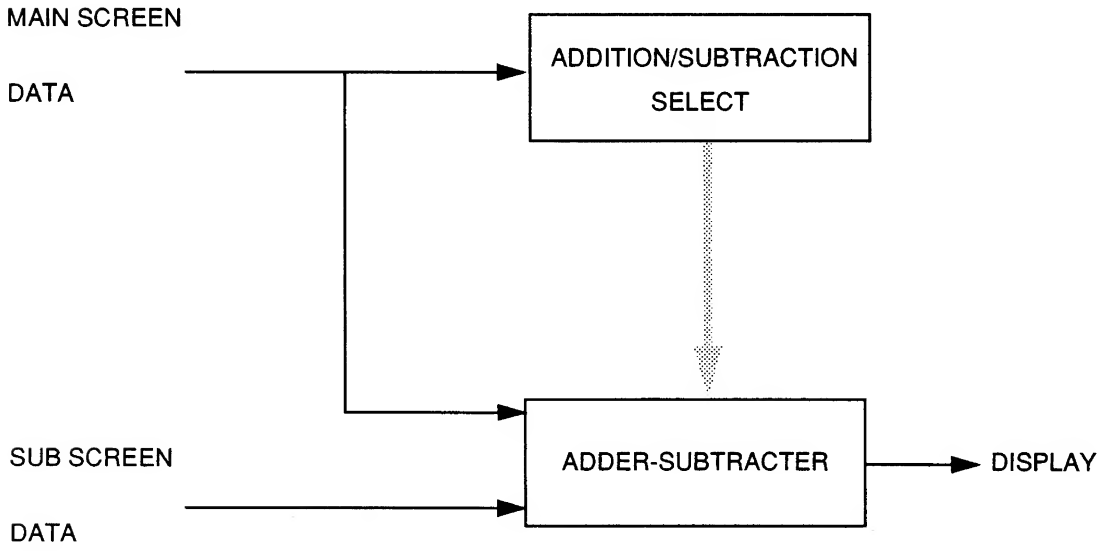
This function is the addition (Overlapping Light) or the subtraction (Lens Filter) for the main screen and the sub screen in order to have the effect of transparency.

7.1.2 FUNCTION

This function displays the result after the addition or subtraction of RGB data on the main screen and sub screen. This function can also select BG screen or OBJ data on the main screen to be added to or subtracted from the sub screen, similar to the figure below. However, when there is no screen data on the sub screen (screen is clear), the color constant explained on page 1-7-4 will be added or subtracted.

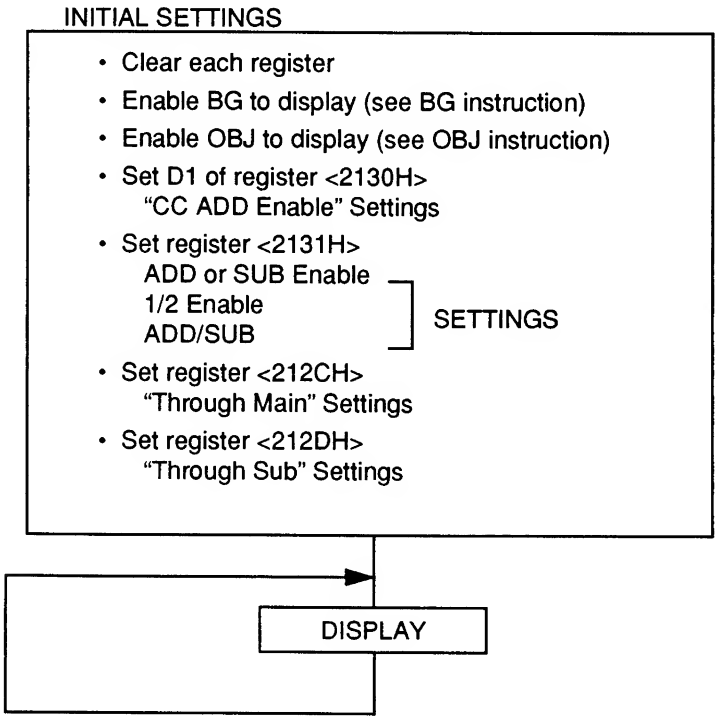
When the result of addition or subtraction exceeds 31, the value becomes 31. When the result of addition or subtraction is less than 0, the value becomes 0.

Please do not use this function on BG mode 5 or 6.



(NCL PG 13)

7.1.3 SETTING EXAMPLE



NOTE: When the main screen data is the OBJ, it will be added to or subtracted from the sub screen data only for the OBJ of the pallet code (4 to 7).

NOTE: When "1/2 Enable" of register <2131H> is enabled, the addition/ subtraction result of each RGB becomes 1/2.

7.2 COLOR CONSTANT ADDITION/SUBTRACTION

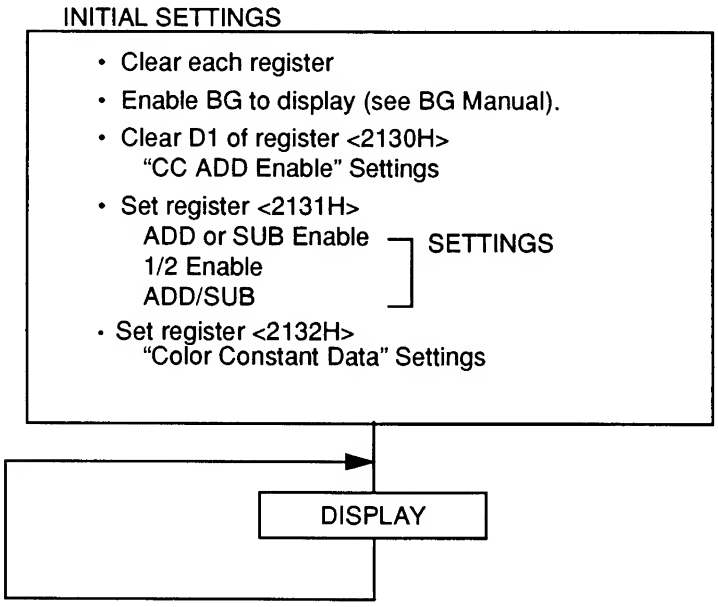
7.2.1 OUTLINE

This function can perform addition (overlapped light) or subtraction (lens filter) with RGB value (color constant) set by the main screen and register <2132H>. This will change the color on the display area.

7.2.2 FUNCTION

This function can perform addition/subtraction by using the RGB value (color constant) which is set by register <2132H> instead of the sub screen of the addition/subtraction screen described previously.

7.2.3 SETTING EXAMPLE



7.3 COLOR WINDOW (Combination of Window & Addition/ Subtraction)

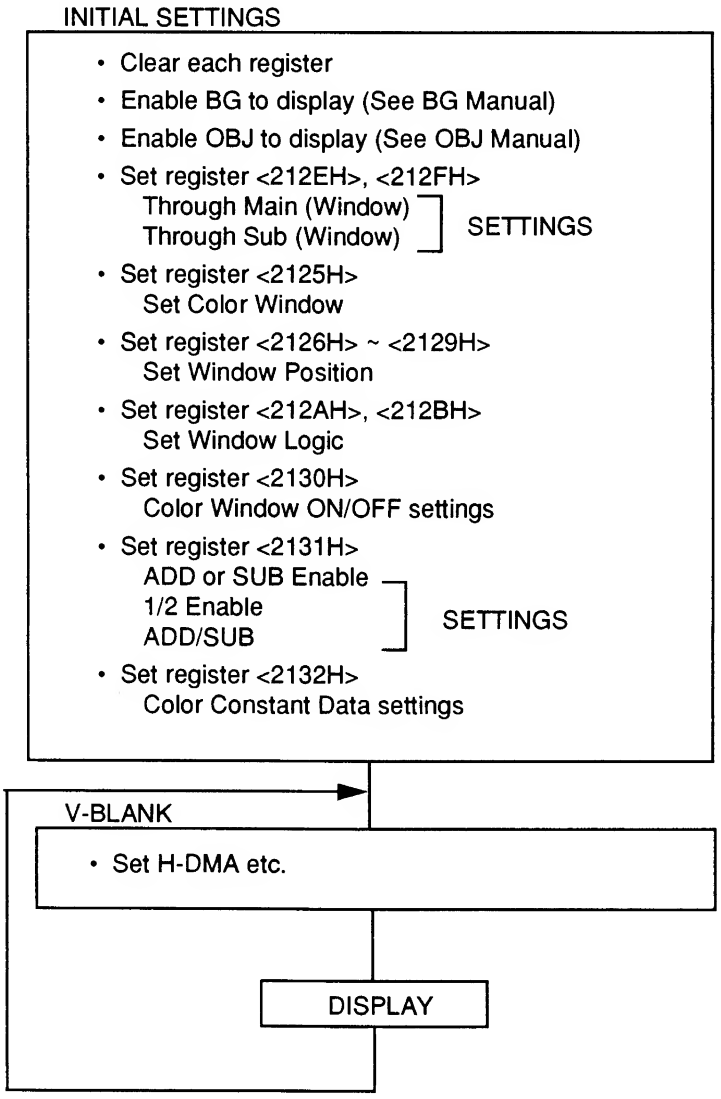
7.3.1 OUTLINE

The Screen Addition/Subtraction or the Color Constant Addition/Subtraction can be performed inside or outside the window (only one or the other).

7.3.2 FUNCTION

This function can select what portion of the window should be displayed and added or subtracted on each main screen and sub screen. The following is the function of window, the screen addition/subtraction and the color constant addition/subtraction.

7.3.3 SETTING EXAMPLE



(NCL PG 16)

Chapter 8. CG Direct Select

8.1 OUTLINE

On BG-1 in Mode 3, 4 and 7, the character data can be used as the color data without using CG-RAM color data. BG-1 can be displayed using 2048 colors on Mode 3 and 4, and 256 fixed colors on Mode 7. BG-2 and OBJ can use the CG-RAM color data without being limited to the color data on BG-1,

8.2 FUNCTION

When BG-1 on Mode 3, 4 and 7 is displayed on the TV screen, this function will display 8-bit color data per character dot without using the CG-RAM. The CG-RAM data is used for the objects and other background screens.

8.3 SETTING EXAMPLE

- Enable BG to display (See BG Instruction)
- Set "D0" of register <2130H>
  "Direct Select" Settings

NOTE: See page A-17 for color data.

Chapter 9. H-Pseudo 512

9.1 OUTLINE

In modes other than 5 and 6, this function provides gradation between 2 dots which are next to each other horizontally, which changes the color smoothly.

9.2 FUNCTION

This function utilizes screen addition/subtraction. The color constant addition/subtraction can not be done at the same time that this function is performed.

9.3 SETTING EXAMPLE

- Enable BG to display (see BG instruction)
- Set "D3" of register <2133H>
  "Pseudo 512" settings
- Set register <212CH>, <212DH>
  Through Main ┐
  Through Sub ┘ SETTINGS
- Set D1 of register <2130H>
  "CC ADD Enable" settings
- Set register <2131H>
  ADD or SUB Enable
  1/2 Enable ┐
  ADD/SUB ┘ SETTINGS

(NCL PG 18)

Chapter 10. Complementary Multiplication (Signed Multiplication)

10.1 OUTLINE

The 2's complement multiplication will be performed with high speed. For example, to calculate the rotation parameter in mode 7, it will lighten the burden of the CPU processing.

10.2 FUNCTION

The high speed multiplication of 16-bit (2's complement) and 8-bit (2's complement) will be performed with "no-wait," and the result becomes 24-bit (2's complement).

10.3 SETTING EXAMPLE

- Set BG other than MODE-7 (or V-Blank/Forced Blank)
  (Except during V-Blank or Forced Blank period)
- Write lower 8-Bit (Multiplicand) to register <211BH>: (Input)
- Write higher 8-Bit (Multiplicand) to register <211BH>: (Input)
- Write register 8-Bit (Multiplier) to register <211CH>: (Input)
- Read register <2134H> ~ <2136H>: (Result)

(NCL PG 19)

Chapter 11. H/V Counter Latch

11.1 OUTLINE

This function is used for synchronization of process timing by tracking the scanning beam on the screen.

11.2 FUNCTION

This function sets the vertical and horizontal counter value (when register <2137H> is read) and tracks the raster beam on the screen by reading the register value. (The scanning is synchronized with an internal vertical and horizontal counter.)

11.3 SETTING EXAMPLE

- Read register <2137H>: (counter latch)
- Read register <213FH>
  (Initialize register <213CH>, <213DH> in the order of Low and High)
- Read register <213CH>, <213DH>

Chapter 12. Offset Change

12.1 OUTLINE

The horizontal and vertical scroll (offset) value can be performed every horizontal 8-dot (character unit) in mode 2, 4, and 6. The other part of the screen can be brought into the middle of the frame in order to have the effect of a window. A partial vertical scroll can also be made.

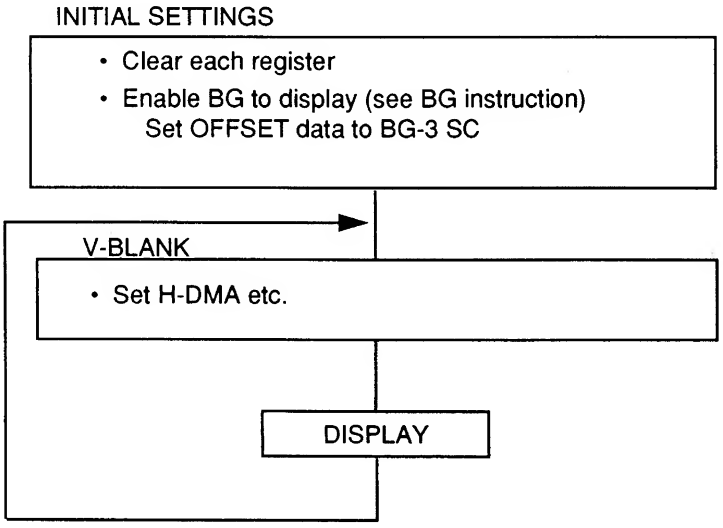
12.2 FUNCTION

This function can be used in any of the three ways, listed below.

- Affect BG-1 only
- Affect BG-2 only
- Affect both BG-1 and BG-2

The offset for both H and V can be changed at every character unit on mode 2 and 6, but the offset for either H or V (only one or the other) can be changed on mode 4. The same offset will be performed on each line once the offset data for a horizontal line (32 characters) is set. To change the setting of the other offset value, depending on the scanning line, change "BG-3 SC Offset Address" or "BG-3 SC Base Address" during the H-DMA period.

12.3 SETTING EXAMPLE



(NCL PG 21)

Chapter 13. Standard Controller

13.1 OUTLINE

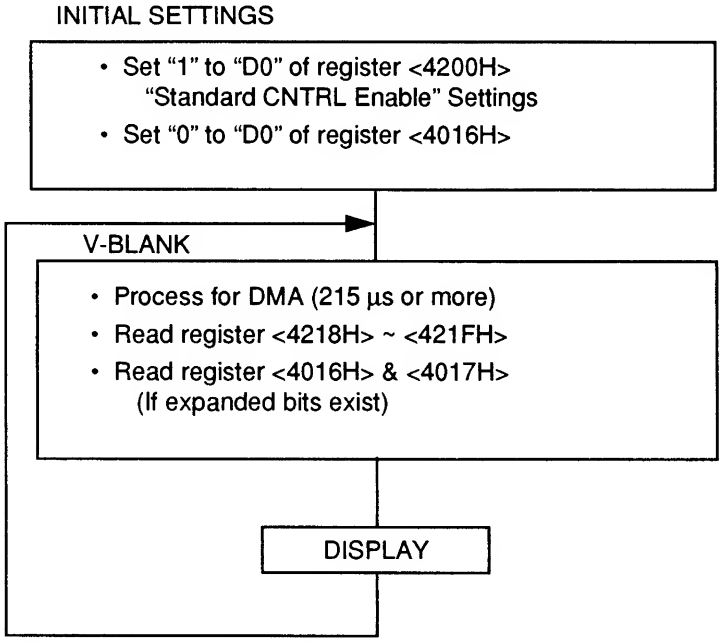
The switch status of the standard controller can be read automatically in serial order and will be converted to parallel data.

13.2 FUNCTION

Two standard controllers can be connected to the Super NES. Four standard controllers may be connected by using an expanded connector, such as MultiPlayer 5 (refer to "Accessories"). Single bit data is assigned to each switch. Up to 16 bits can be read automatically for one standard controller. The expanded bit data can be read 1 bit at a time by the software, as for the NES. The hardware reads the data for about 215 µs after the V Blank flag is set or NMI is applied. During this data read period, the standard controller register cannot be read properly.

- 215 (214.55) µs is equivalent to 3.4 (3.38) scanning lines; a period of 580 (576) bytes to be transferred by DMA. (If the CPU clock is 2.68 MHz, it is equivalent to 580 machine cycles.) As soon as V-Blank starts, normal flow is to perform general purpose DMA. Therefore, it is convenient if the total number of bytes to be transferred by general purpose DMA is used for read timing. (Please refer to the System Flowchart.)
- The standard controller data (register) should be read after confirming that "JOY-C Enable" of register <4212H> is not set during the V-Blank period, so that valid data can be read.
- After the 18 µs (48 machine cycles with 2.68 MHz) from the beginning of V-Blank, the hardware will start to read. "Standard CNTRL Enable" of register <4212H> cannot be set during this period.

13.3 SETTING EXAMPLE



(NCL PG 23)

Chapter 14. Programmable I/O Port

14.1 OUTLINE

An 8 bit programmable I/O port is provided for interface to peripheral devices, such as; a keyboard, the 3D glass, etc.

14.2 HOW TO USE

A "1" should be written to register <4201H> for the bit to be used as the in-port. The selected bit will become the in-port, which can be read by register <4213H>. Output data should be written to the bit of register <4201H> to be used as the Out-port. This data can be output directly.

★Only 2 of the 8 bits can be used at the connector for the controller (Refer to page 1-28-1).

Chapter 15. Absolute Multiplication/Division

15.1 OUTLINE

Absolute multiplication (8 bit by 8 bit) and absolute division (16 bit by 8 bit) can be done using this function. It is also convenient for processing arrays of tables and can improve the processing speed for multiplication and division.

15.2 FUNCTION

The multiplication calculation between the multiplicand of an 8 bit absolute value (0 ~ 255) and the multiplier of an 8 bit absolute value (0 ~ 255) can be performed and can provide the result of a16 bit product (0 ~ 65025). The division calculation between the dividend of a 16 bit absolute value (0 ~ 65535) and the divisor of an 8 bit absolute value (0 ~ 255) can be performed and can provide the result of a 16 bit quotient (0 ~ 65535) and a 16 bit remainder.
If the divisor is "0" in the division calculation, the quotient value becomes 65535 (0FFFFH) and the remainder becomes the dividend value. Therefore, caution is required.
It takes about 8 machine cycles for the multiplication calculation and about 16 machine cycles for the division calculation. The register value for multiplicand and dividend will not be destroyed even after the operation.

15.3 SETTING EXAMPLE

- In case of Multiplication
  - Set register <4202H>
    "Multiplicand-A" Settings
  - Set register <4203H>
    "Multiplier-B" Settings
  - Wait for 8 Machine Cycles
  - Read register <4216H>, <4217H>
    Read Product-C
- In case of Division
  - Set register <4204H>, <4205H>
    "Dividend-C" Settings
  - Set register <4206H>
    "Divisor-B" Settings
  - Wait for 16 Machine Cycles
  - Read register <4214H>, <4215H>
    Read Quotient-A
  - Read register <4216H>, <4217H>
    Read Remainder

Chapter 16. H/V Count Timer

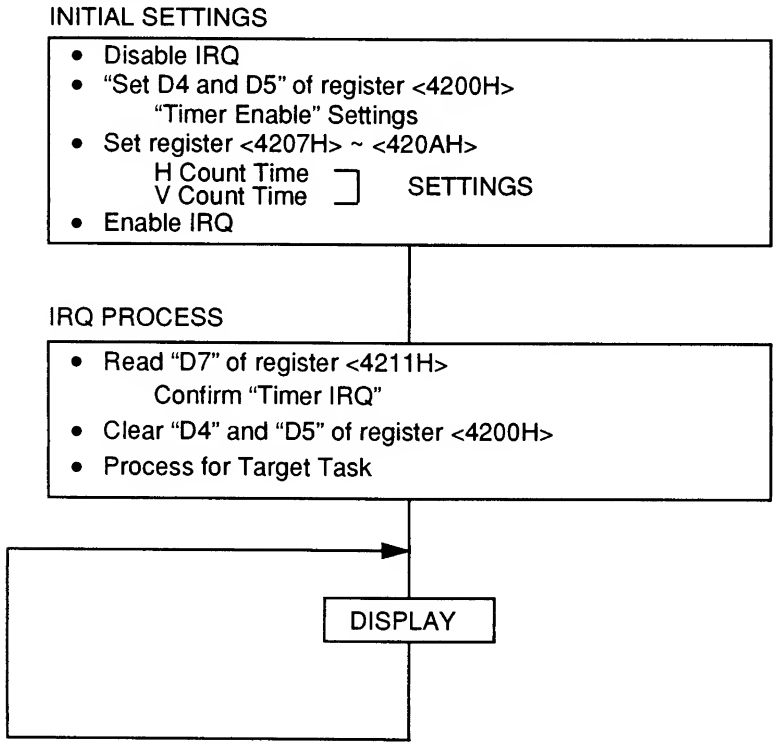
16.1 OUTLINE

The Super NES has a timer synchronizing with the display on the TV screen, which is used for adjusting the synchronization of the scanning process on the screen and software execution.

16.2 FUNCTION

This function can generate the interrupt at either a V or H position of the scanning lines. It can also generate the interrupt at any position of the scanning line.

16.3 SETTING EXAMPLE



(NCL PG 26)

Chapter 17. Direct Memory Access (DMA)

The DMA is the method to transfer the data in the same manner as the data transfer which is done by the CPU. However, the DMA can transfer the data at high speeds by using the hardware instead of the CPU. The SNES has the exclusive DMA, since the picture data has to be transferred rapidly.
The DMA for the SNES is to transfer the data between "A-Bus Address" in the CPU (0000000 ~ 0FFFFFF) and "B-Bus Address" in the S-PPU (0002100 ~ 00021FF), which has 8 channels total. There are two kinds of DMA: general purpose DMA and H-DMA. Either can be set at each channel. The data can be transferred between the same DMA's in the order of lower channel numbers (0 ~ 7). The H-DMA can interrupt even during the transfer by the general purpose DMA, which means that the H-DMA has higher priority than the general purpose DMA. Furthermore, the CPU process stops automatically during the DMA period, and will start again after the DMA is completed. It is not necessary to observe the DMA completion by the CPU.

17.1 GENERAL PURPOSE DMA

17.1.1 OUTLINE

This function can transfer the data rapidly between 2 types of memory devices: memory which can be accessed directly by the CPU, such as a ROM on the game cartridge, and memory which has to be accessed through the S-PPU, such as the V-RAM.

17.1.2 FUNCTION

The maximum area of the A-Bus address which can be used in one channel is limited in one bank (65,536 Byte). Therefore, in case of spreading over more than 2 banks, it is necessary to use more than 2 channels or transfer twice. One A-Bus address basically is increased every time 1 byte of data is transferred. However, it can be decreased or fixed depending on the settings ("d3" and "d4" of register <43X0H>).

The following table shows four types of B-Bus address changes:

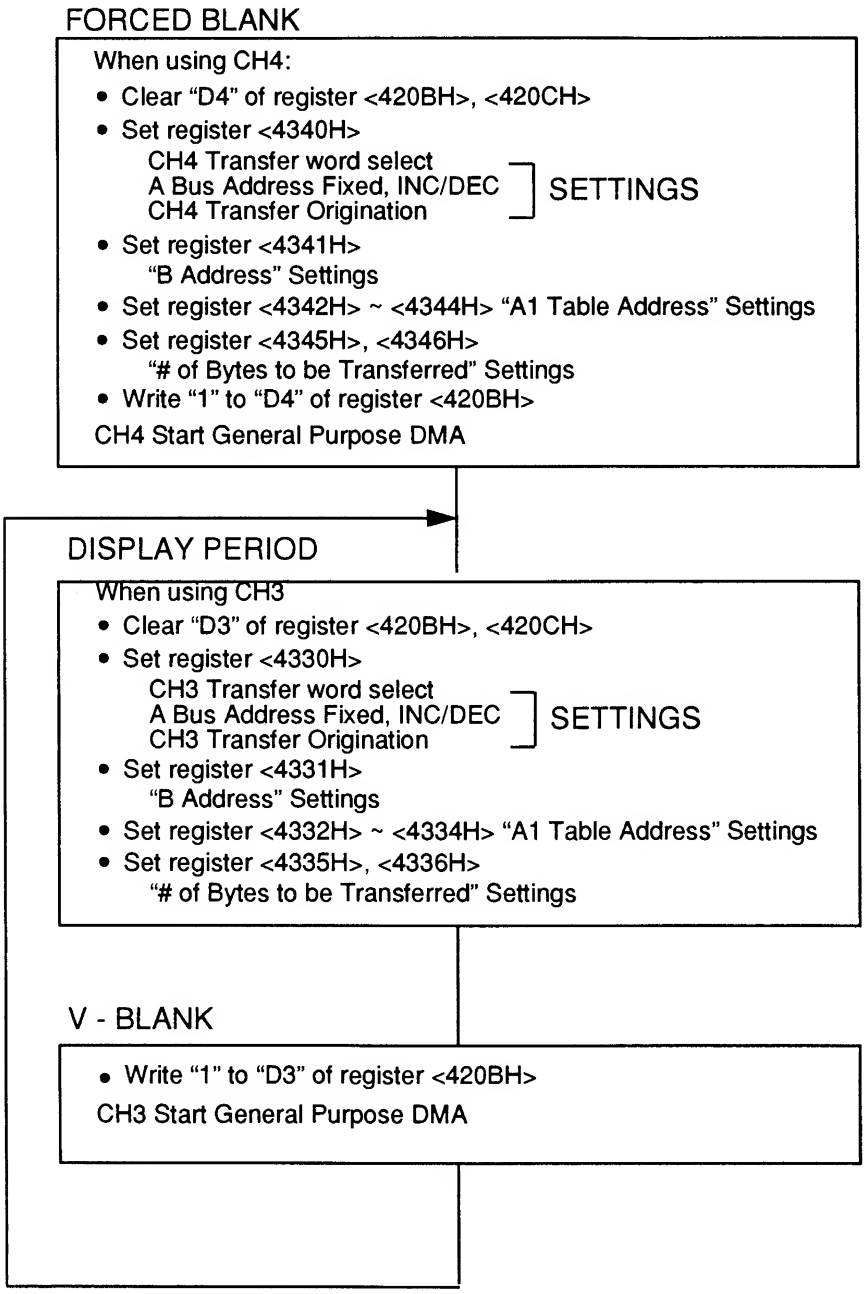
Table 2-17-1 B-Bus Address Changes

Transfer Word Select <43X0H> / # of Transfer (# of Byte)	D2 ~ D0 000 or 010	D2 ~ D0 001	D2 ~ D0 011	D2 ~ D0 100
0	B	B	B	B
1	B	B + 1	B	B + 1
2	B	B	B + 1	B + 2
3	B	B + 1	B + 1	B + 3
4	B	B	B	B
5	B	B + 1	B	B + 1
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

- In case of 224 lines, general purpose DMA can transfer 6K byte data maximum during V-Blank period.

NOTE: B means the data of register <43X1H>

17.1.3 SETTING EXAMPLE



(NCL PG 29)

17.2 H-DMA

17.2.1 OUTLINE

This is a special DMA which can transfer data automatically, synchronizing with the H-Blank. The S-PPU settings can be varied by each horizontal scan line and special effects can be added to the picture.

17.2.2 FUNCTION

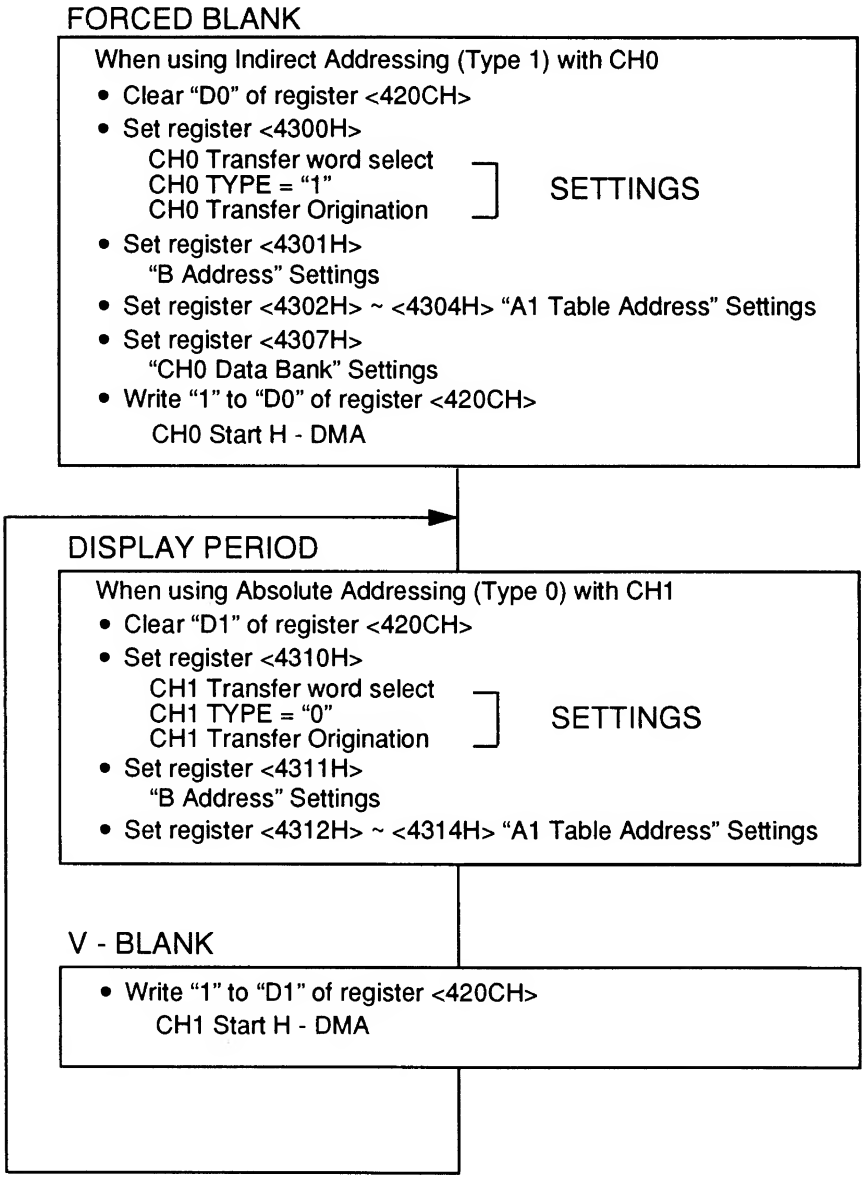
This function transfers the data from the A-Bus memory (CPU memory) to the S-PPU register. There are two kinds of addressing modes on the A-Bus side; absolute and indirect addressing. Either type of addressing can be set by each channel. There are two kinds of data transfer. One is to transfer a set of data during each horizontal blanking period. The other is to transfer a set of data every certain number of horizontal blanks.

Table 2-17-2 B-Bus Address Change

Transfer Word Select <43X0H> / # of Line to be transferred	D2 ~ D0 000	D2 ~ D0 001	D2 ~ D0 010	D2 ~ D0 011	D2 ~ D0 100
1	B	B	B	B	B
				B	B + 1
		B + 1	B	B + 1	B + 2
				B + 1	B + 3
2	B	B	B	B	B
				B	B + 1
		B + 1	B	B + 1	B + 2
				B + 1	B + 3
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•

NOTE: B means the data of register <43X1H>.

17.3 SETTING EXAMPLE



(NCL PG 31)

Chapter 18. Interlace

18.1 BG MODE 0 ~ 4 & 7

When "1" is written to D0 of register <2133H>, the picture signal output from the Super NES will be the interlace signal. In the case of BG modes 0 through 4 and 7, the same picture will be displayed unless the picture data is changed between the 1st field and the 2nd field. (Refer to BG Screen in Appendix A.)

18.2 BG MODE 5 & 6

When using interlace on BG mode 5 and 6, the vertical resolution will be doubled in appearance. The picture is displayed using a one frame combination of the 1st field and 2nd field. (Refer to BG Screen in Appendix A.)

18.3 OBJ

When "1" is written to "D1" of register <2133H>, the vertical resolution will be doubled as in the case of BG Mode 5 and 6, because a picture is generated using one frame. The range of the V-position for OBJ is 0 through 255 and this range will not be doubled.

Chapter 19. H-512 Mode (BG Mode 5 & 6)

19.1 MAIN SCREEN & SUB SCREEN SETTINGS

The screen addition/subtraction function should not be used, because a part of both main screen and sub screen functions are used in this mode. With the exception of color constant addition/subtraction, "1" should be written to D4 and D5 of register <2130H> and the sub-switch should be off. The same data should be written to registers <212CH>, <212DH>, <212EH>, and <212FH>. "Through" should be the same for both the main and sub-screens.

19.2 FIXED COLOR ADDITION/SUBTRACTION

D0 ~ D5 of register <2131H> is a flag which can select the main screen for addition/subtraction. Because a part of both main screen and sub screen functions are used, this selection cannot be performed. It is necessary to write "1" to 6 flags (D0 ~ D5) when color constant addition/subtraction is performed. The remaining settings are the same as the normal Color Constant Addition/Subtraction. There will be addition/subtraction every 2 dots, horizontally, in the color window function, because the window has only 256 positions horizontally.

19.3 DISPLAY WITH OBJ

The name H-512 indicates a horizontal resolution of 512 for BG. The horizontal resolution for the OBJ is only 256-dot, regardless of the BG mode. The priority order for BG is determined by every dot.

19.4 OTHERS

See "BG Screen" in the Tables of Appendix for details.

Chapter 20. OBJ 33's Lines Over & Priority Order

20.1 33'S RANGE OVER

The number of OBJs which can be displayed in a horizontal line is limited. One of these limitations is called the "33's Range Over." This limits the number of OBJs which can be displayed in a horizontal line, regardless of the OBJ size. If "33's Range Over" has occurred in one field (at least one line), "D6" of register <213EH> will be set. For the line in which this "33's Range Over" occurs, only 32 OBJs can be displayed out of 33 or more OBJs present. The 32 OBJs displayed are selected using the priority order (selected from smaller OBJ number).

NOTE: "The number of displayed OBJs" counts OBJs hidden by BG window or other OBJs.

NOTE: If H-position is minus, and the OBJ is not displayed on the screen area (located on the left of the screen to be displayed), "the number of displayed OBJs" does not count them.

20.2 35'S TIME OVER

The other limitation on the horizontal line is called "35's time over." This limits the number of OBJs (converted to character size 8-dot x 8-dot) that can be displayed. If the "35's Time Over" has occurred in one field (at least one line), "D7" of the register <213EH> will be set. In the line in which this "35's Time Over" has occurred, only 32 of the total OBJs available can be displayed according to the priority order (selected from larger OBJ number). This limit is due to a conversion limit of less than 35 OBJs (8 x 8) displayed per horizontal line. "These 32 OBJs must satisfy the display condition explained in "33's Range Over", above.

NOTE: There are characters (8-dot x 8-dot) which are not displayed on the display area depending on OBJ size and position. But they are not included in this limitation (34 or less).

20.3 PRIORITY ORDER SHIFTING

As mentioned above, limited numbers of OBJs can be displayed in a line and are related to the priority order. It is desirable to develop a game within this limitation. However, sometimes OBJs need to be displayed beyond this limitation. This can be accomplished using virtual OBJs. One method is to change the priority order every frame. Another method changes the OBJ data order through programming. The Super NES also contains a function which rotates the priority order of 128 OBJs. When using these methods, consider that the OBJ will flash every frame unit and the priority order among OBJs will change. The method for assignment is as follows:

Step 1. Display the OBJ.

Step 2. Write "1" to "D7" of register <2103H>.

Step 3. Write the highest priority OBJ number (0 ~ 127) to "D1 ~ D7" of register <2102H> during V-Blank period every frame.

Step 4. Repeat step 3.
When OBJ number stored in step 3 is "n".

OBJ NUMBER	PRIORITY ORDER
OBJ 0	129 - n
•	•
•	•
•	•
OBJ (n - 1)	128
OBJ (n)	1
OBJ (n + 1)	2
•	•
•	•
•	•
OBJ 127	128 - n

(NCL PG 35)

Chapter 21. CPU Clock and Memory Mapping

21.1 CPU CLOCK

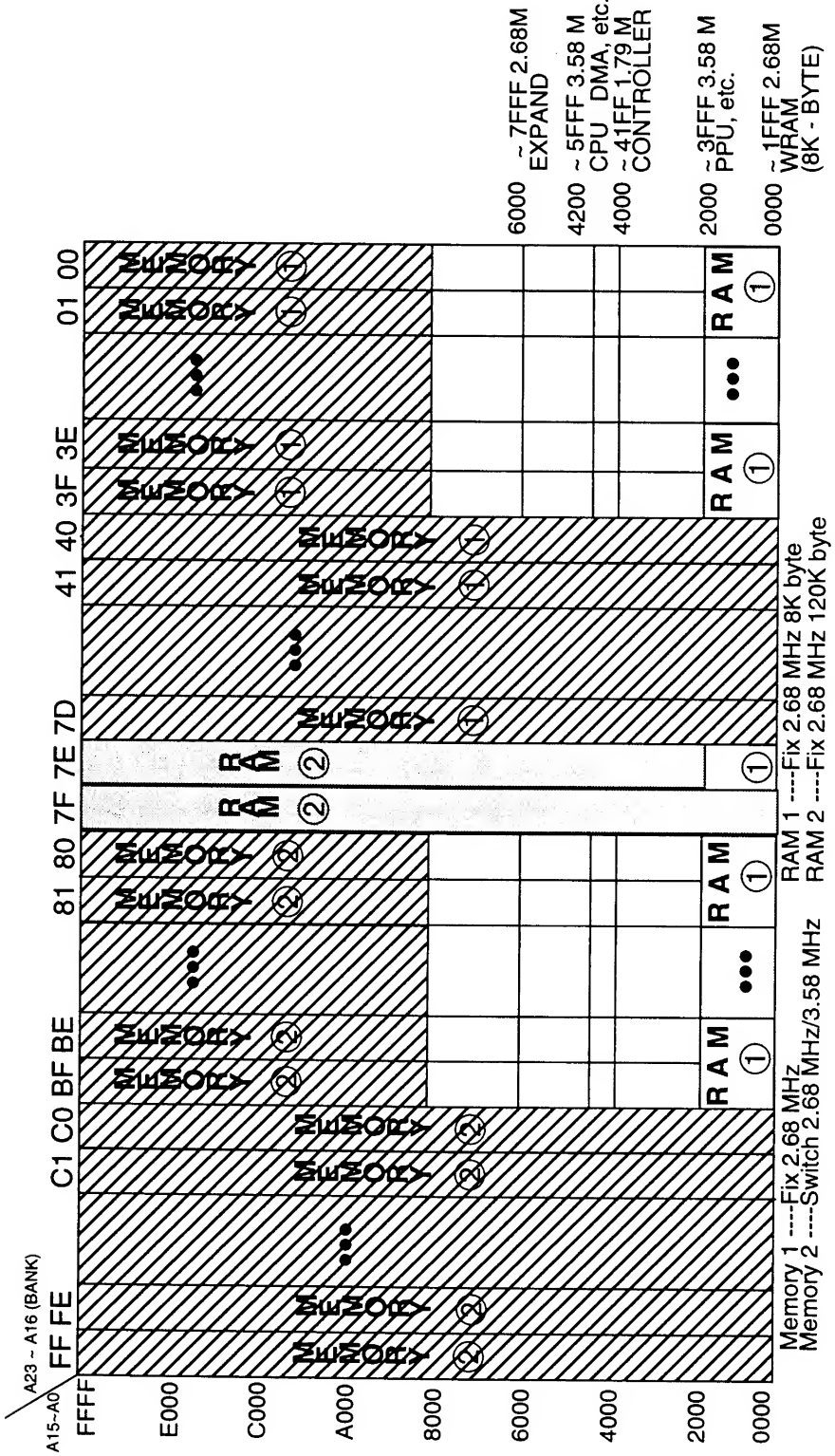
The CPU clock can be switched automatically, depending on the address to be accessed by the CPU. Three clock speeds are available: 3.58MHz, 2.68MHz, and 1.79 MHz. The device speed (ROM, RAM, LSI, etc.) will determine the speed to be used. If a medium speed ROM and RAM (access time less than 200ns) are used in the cartridge, it will be mapped to the address area for 2.68MHz. If high speed (access time less than 120ns) are used, it will be mapped to the address area for 3.58MHz. Please refer to "Frequency & Address Mapping" for the relation between the address and the clock. Two clocks (2.68MHz & 3.58MHz) can be selected by setting D0 of register <420DH> for the range of memory "②" shown in the illustration on the next page. The default setting is 2.68MHz. The CPU is operated internally with a 3.58MHz clock speed. (Regardless of the address, DMA will be performed with 2.68MHz clock speed).

21.2 CPU MEMORY MAP

Please refer to "Frequency & Address Map" on the next page. The WRAM (8K-Byte) is mapped to address (0000 ~ 1FFF) of banks (00 ~ 3F), (80 ~ BF) and 7E. This is the WRAM used as common bank. This 8K-Bytes can be accessed from any bank described above. The WRAM (120K-Byte) is mapped to address (2000 ~ FFFF) of bank 7E and (0000 ~ FFFF) of bank 7F. Therefore, the WRAM (128K-Byte total) is included in the Super NES unit. This 128k-Byte (RAM ①, RAM ②) is one consecutive memory and can be accessed from the B - Bus address. The address "2000 ~ 5FFF" of bank "00 ~ 3F" and "80 ~ BF" are reserved as a register area of the S-PPU, DMA, etc. Because this basically is reserved as a common bank area, the S-PPU and DMA register can be accessed from any bank above.

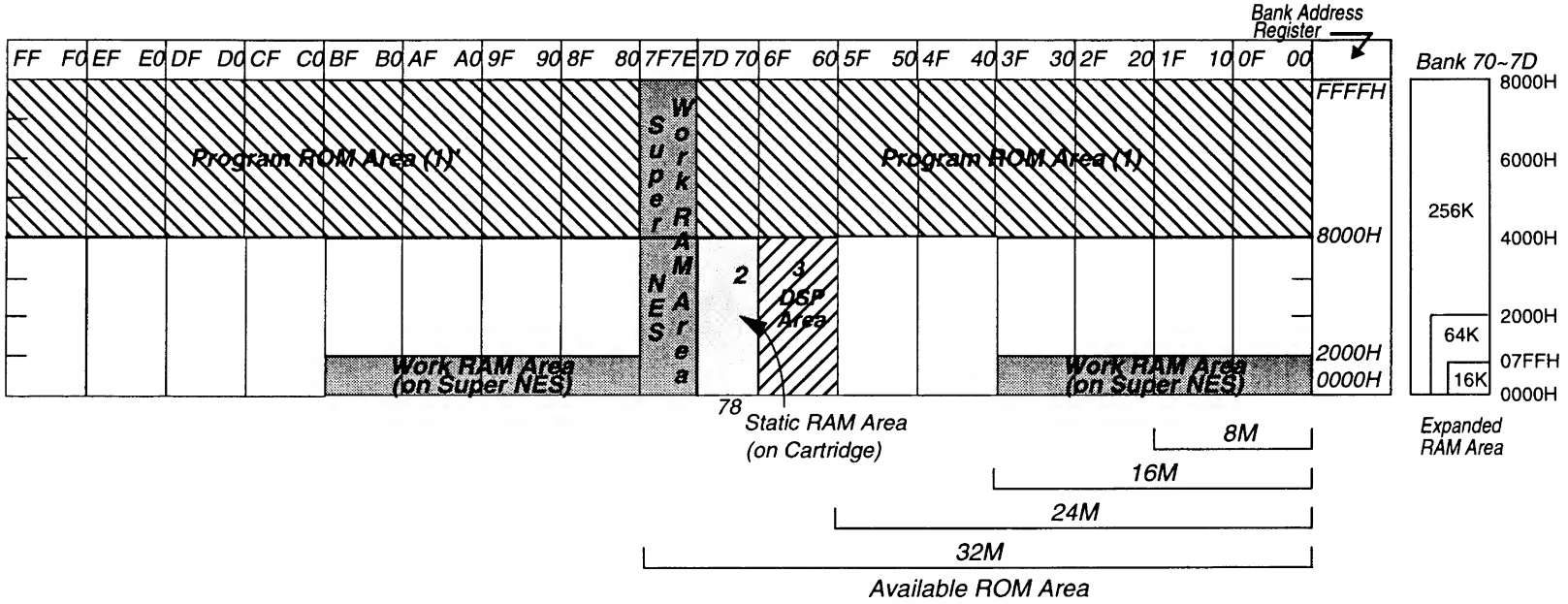
(NCL PG 36)

Figure 2-21-1 Super NES CPU Memory Map



(NCL PG 37)

Figure 2-21-2 Super NES Memory Map (Mode 20)



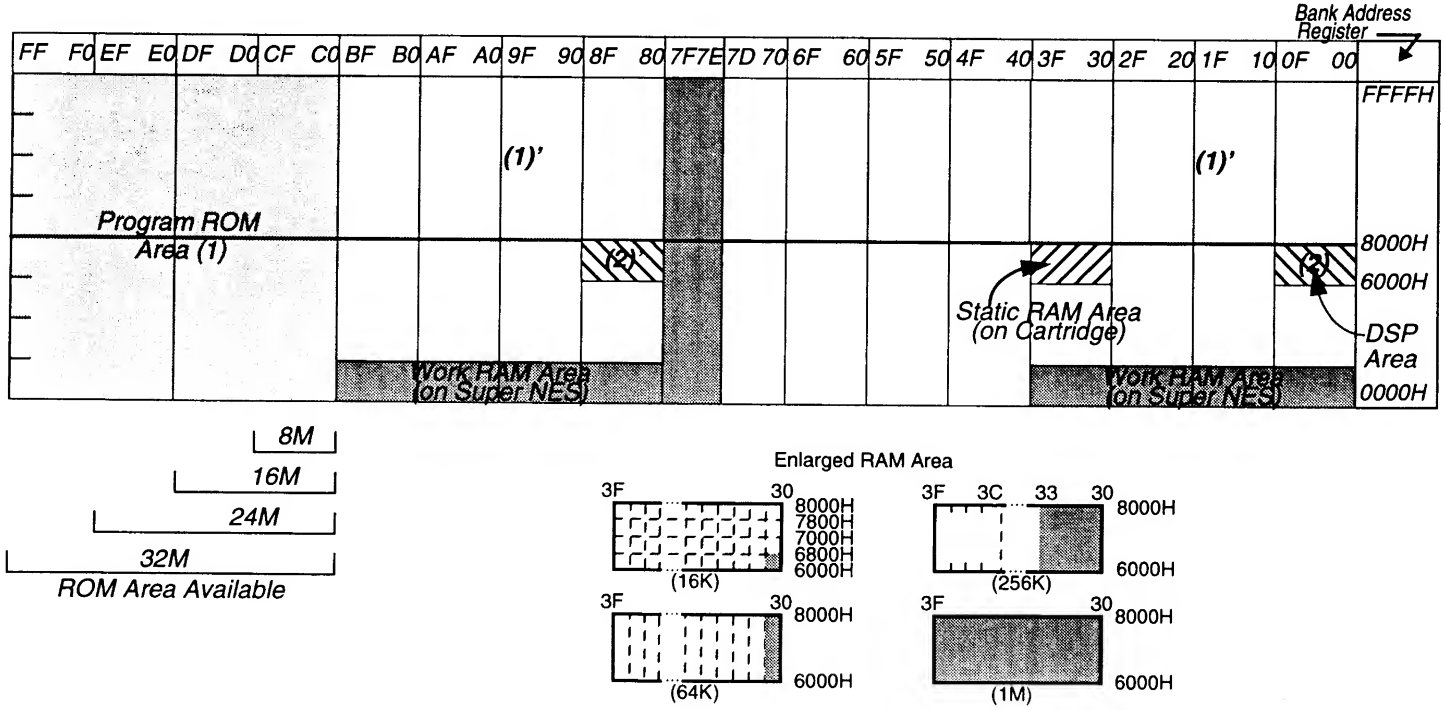
Note 1: The ROM image for bank 00H ~ 7DH is generated in bank 80H ~ FDH.

Note 2: Set start vector and general registration area at address FFC0H ' (ROM address 00:7FC0H) in bank 00H.

Note 3: Programs located in the ROM area of bank 80H ~ FFH can be executed in the high speed mode. Specify the need for the high speed mode in the submission form.

Note 4: When the program ROM is less than 8M, the DSP area is from 8000H ~ FFFFH in bank 30H ~ 3FH.

Figure 2-21-3 Super NES Memory Map (Mode 21)

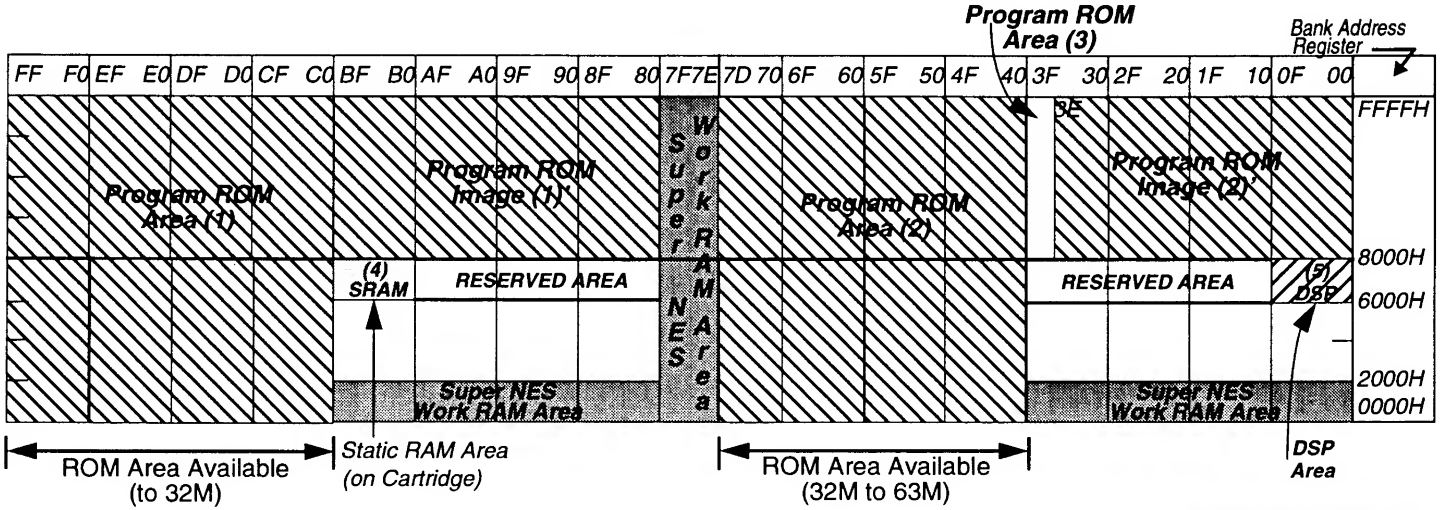


Note 1: In memory Mode 21, memory can be added from bank C0H to bank FFH (maximum is 32M bit). The ROM image of "8000H - FFFFH of bank C0H - FFH" will appear on "bank 00H - 3FH". Set vectors (i.e., Reset Vector) in the vector area of bank C0H".

Note 2: The ROM image from address 8000H ~ FFFFH of bank C0H ~ FFH is generated in bank 00H ~ 3FH and bank 80H ~ BFH.

Note 3: Programs located in the area of bank 80H ~ FFH can be executed in the high speed mode. Specify the need for the high speed mode in the submission form.

Figure 2-21-4 Super NES Memory Map (Mode 25, ROM Size Greater than 32 Mbits only)

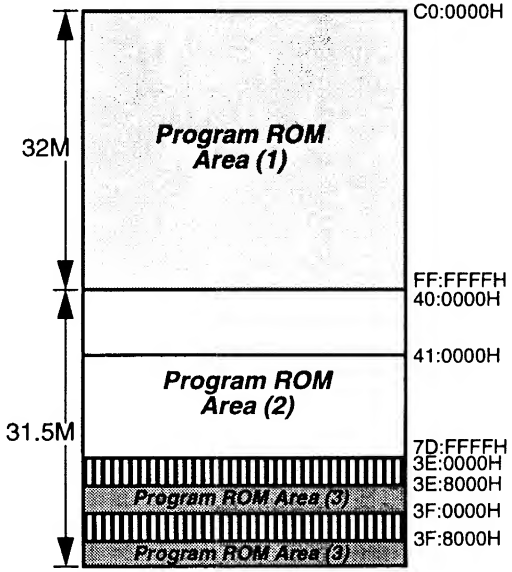


Note 1: The ROM image of 8000H~FFFFH of bank 40H~7DH will appear at bank 00H~3DH. Set vectors and registration data in FFB0H of bank 40H (ROM address 40:FFB0H).

Note 2: Programs located in the area of bank 80H~FFH can be executed in the high speed mode (3.58 MHz).

Note 3: Don't access null area.

Note 4: Use the area of bank 3E and 3F as the program ROM area of bank 7E and 7F.



Chapter 22. Super NES Functional Operation

This chapter provides the user with a basic understanding of the functional purpose for each of the major components of the Super NES control deck. Refer to the Super NES Functional Block Diagram (opposite page) while reading the following paragraphs.

22.1 SUPER NES CPU

This is the Central Processing Unit for the Super NES. It coordinates all functions of the Super NES control deck and peripheral devices which are attached to the Super NES.

22.2 SUPER NES PPU1 AND PPU2

These 2 units work together as the Picture Processing Unit for the Super NES. Pictures are generated for display based upon control inputs from the Super NES CPU. In general, PPU1 is used to generate background character data, rotation, and scaling; while PPU2 performs special effects like windows, mosaic, and fades.

22.3 SUPER NES WRAM

The work RAM (WRAM) is a custom 128K x 8 bit RAM used by the Super NES CPU for data storage. Direct Memory Addressing (DMA) can be used by the Super NES CPU for rapid bulk transfer of data.

22.4 VRAM

The VRAM is composed of 2 - 32K x 8 bit S-RAMs. This unit is used by PPU1 to store background character data until needed for display.

22.5 AUDIO PROCESSING UNIT (APU)

The Audio Processing Unit performs all sound functions for the Super NES and is composed of the following units.

22.5.1 SOUND CPU

The Sound CPU is the central processing unit for the Super NES Audio Processing Unit. It controls sound functions much in the same way that the Super NES CPU controls functions of the Super NES.

22.5.2 SOUND DSP

The DSP has 8 channels of pulse code modulated (PCM) sound, a noise generator, echo, sweep, envelope, and other circuits to reproduce tone qualities from RAM data.

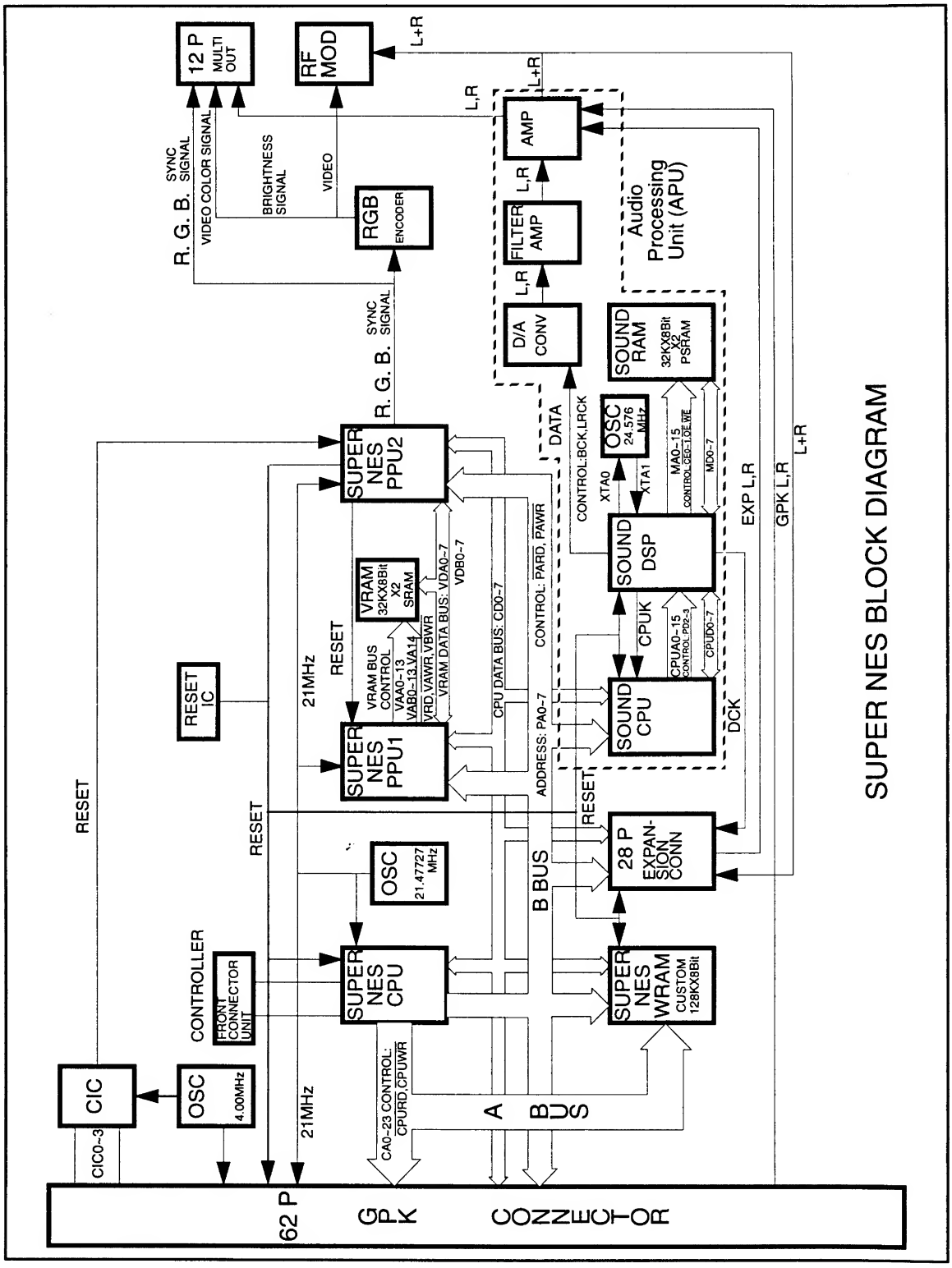


Figure 2-22-1 Super NES Functional Block Diagram

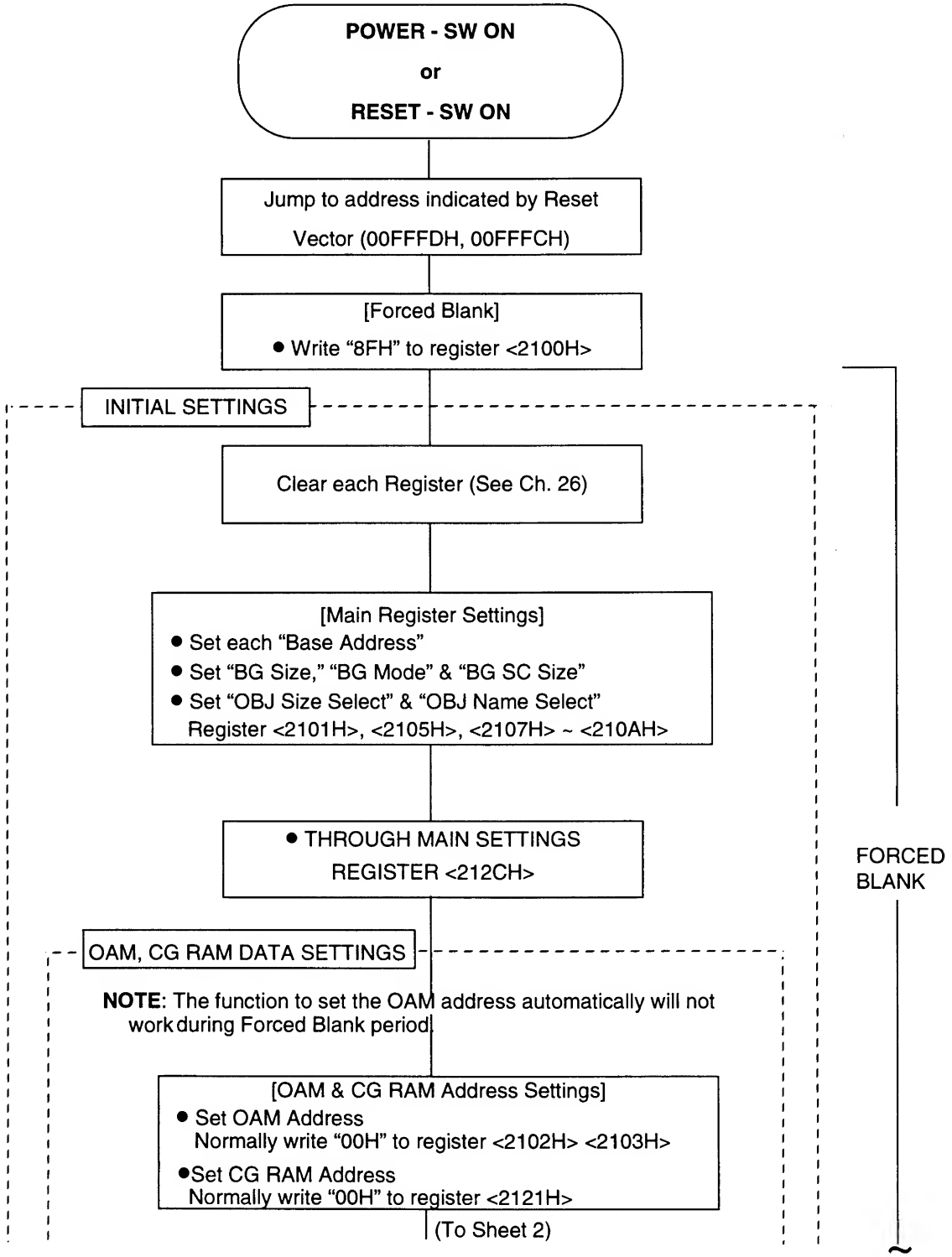
22.5.3 SOUND RAM

The Sound RAM is composed of 2-32Kx8 bit SRAMs. Program and tone data are loaded from the game pak to the sound RAM by the Sound CPU. The RAM is time shared by the Sound CPU and DSP.

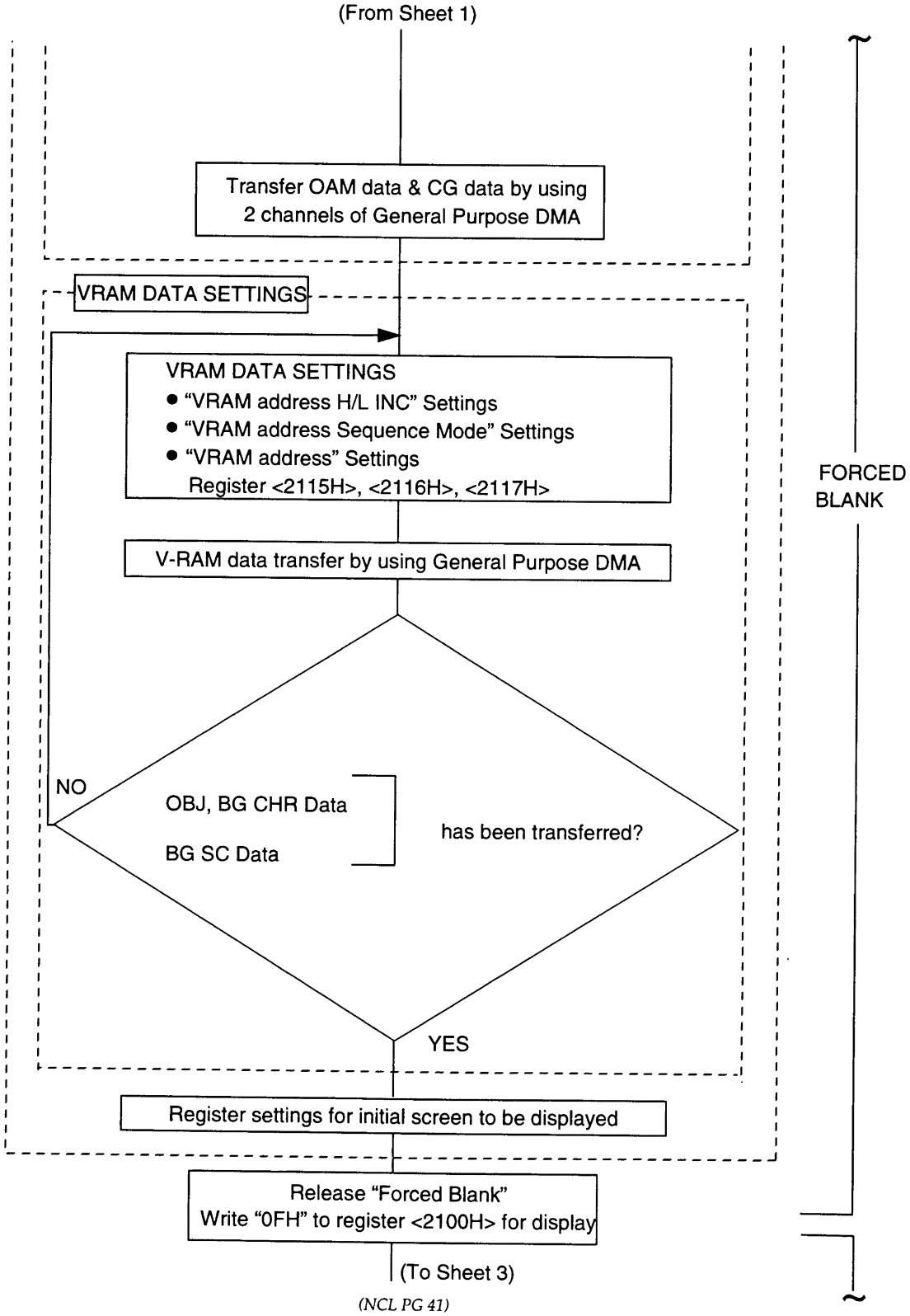
22.5.4 D/A CONVERTER

Converts the digitized sound to an analog signal which is filtered and amplified to produce the L+R (mono) output through the RF Modulator and L,R (stereo) outputs through the multi-out connector.

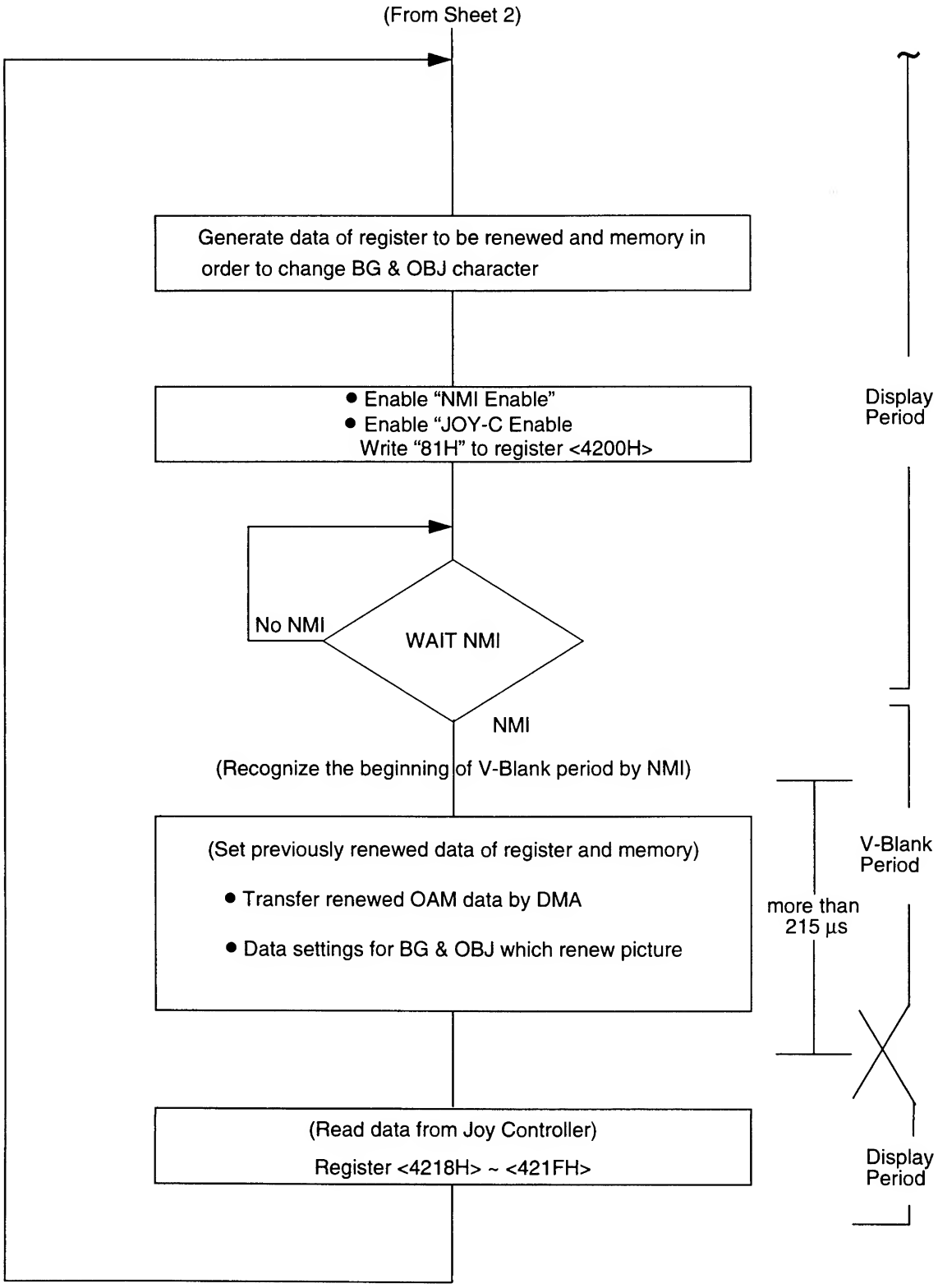
Chapter 23. System Flowchart



(NCL PG 40)


(From Sheet 1)
Transfer OAM data & CG data by using 2 channels of General Purpose DMA
VRAM DATA SETTINGS
VRAM DATA SETTINGS
● "VRAM address H/L INC" Settings
● "VRAM address Sequence Mode" Settings
● "VRAM address" Settings
Register <2115H>, <2116H>, <2117H>
V-RAM data transfer by using General Purpose DMA
NO
OBJ, BG CHR Data
BG SC Data
has been transferred?
YES
Register settings for initial screen to be displayed
Release "Forced Blank"
Write "0FH" to register <2100H> for display
(To Sheet 3)
FORCED BLANK

(NCL PG 41)


(From Sheet 2)
Generate data of register to be renewed and memory in order to change BG & OBJ character
• Enable "NMI Enable"
• Enable "JOY-C Enable
Write "81H" to register <4200H>
No NMI
WAIT NMI
NMI
(Recognize the beginning of V-Blank period by NMI)
(Set previously renewed data of register and memory)
• Transfer renewed OAM data by DMA
• Data settings for BG & OBJ which renew picture
(Read data from Joy Controller)
Register <4218H> ~ <421FH>
Display Period
V-Blank Period
more than 215 µs
Display Period

(NCL PG 42)

Chapter 24. Programming Cautions

24.1 CAUTION #1

Registers <210DH> ~ <2114H> and <211BH> ~ <2120H> must be accessed in the order of Low and High twice (Read Twice or Write Twice). If it is not known whether the next access should be low or high, initialize as follows:

OAM, CGRAM, VRAM	Set the address again.
Other Registers (Write)	The lower data should be written more than one time, and the higher data should be written.
H/V Counter Read	The H/V counter will be initialized when the status register <213FH> is read. The data should be read in the order of Low and High.

24.2 CAUTION #2

The period which can be accessed for the register is as follows:

V-RAM, OAM	Forced Blank or V-Blank period only.
CG-RAM	Forced Blank, V-Blank or H-Blank period only.
Other Register (Write)	All period (however, when writing the data, the picture may not be displayed properly).
Other Register (Read)	All period (However, the data which may be changed during display period may not be read properly).

24.3 CAUTION #3

The address space for the V-RAM is 64K-word (1 word = 16-bit) maximum. 32K-word memory is installed in the Super NES unit.

24.4 CAUTION #4

When the V-RAM is accessed from the CPU, the address counter will be increased automatically. For the V-RAM increment mode, please use the register mode designated by the instruction.

24.5 CAUTION #5

When the V-RAM is read continuously, the first address will not be incremented once the V-RAM data has been stored. The first address should be read as dummy data on subsequent passes.

24.6 CAUTION #6

The top color data of each CG color data palette is transparent. Because transparent is a color which is not displayed, any color can be set. The color data of CG address (00H) is normally black (background).

24.7 CAUTION #7

Even though 9-bits are provided as the OAM H-position, the value (100H) must not be used.

24.8 CAUTION #8

Before processing the controller keys, verify which devices are currently connected to the controller ports. The valid identification codes are:

- Standard Controller 0000B
- Super NES Mouse 0001B
- Super Scope 1111B

These codes may be found in bits D3 ~ D0 of registers <4218H> and <421AH>. If the standard controller is used for the game, inputs should be ignored whenever the ID code is not 0000B.

24.9 CAUTION #9

The initial value of the work RAM in the main computer is not set when power is applied to the computer. Programming should be done in such a way that no errors occur when the data is indeterministic. The initial value is different depending upon the computer used. Initialize the entire RAM area when, for example, it has been programmed under the misconception that the data is a fixed value, 00•FF.

24.10 CAUTION #10

When using the battery back-up SRAM, avoid program errors due to data loss. The CPU may crash if the user hits the control deck when the game pak is in use, if the game pak is not inserted properly, or if the game pak connector is dirty. Data loss may be unavoidable in some cases. Before reusing SRAM data, determine if the data is recoverable. One method of detection is to save the data in several areas of the SRAM and calculate the check sums of each area. Before utilizing any data in the SRAM, the program must compare each of the check sums. If the check sums are not equal, the data is corrupted.

(NCL PG 43)

24.11 CAUTION #11

In addition to using a check code to check a hot/cold start, determine if the content of the work RAM used is correct after the reset. Data in work RAM is lost gradually after the power is turned off. The speed at which data is lost differs according to the area. If the device is turned on immediately after it has been turned off, the area that is checked for hot/cold start code may contain the original data. This does not mean that the entire data have been recovered. Guidelines for prevention of data loss are the same as those for the previous caution.

24.12 CAUTION #12

When executing critical commands using the controller keys, such as; modify, erase data, or software reset, use all 16 bits of data including the input device's signature. Corrupt data may be sent by the controller if the controller is unplugged during a game. When the computer is reset using start, select, L, and R controller data simultaneously, verify that:

- The start, select, L, and R are pressed,
- No other keys are pressed, and
- The signature data is 0000.

In other words, check that the key data is 3030H.

24.13 CAUTION #13

Do not place critical game characters within two characters of the perimeter of the display screen area. This area of the television varies from one brand or model to the next. The Super NES may not be able to display characters in some areas if programmed too close to the edge of the screen. Critical game characters include score data and various parameters.

24.14 CAUTION #14

Ensure that the program clears the emulation bit on reset or start-up before executing 65816 instructions (i.e., JMP $808007). This is demonstrated in the programming example, below:

Example:

```
RESET                   ;Reset vector
        SEI             ;Disable interrupt
        CLC             ;Clear carry
        XCE             ;Exchange carry with E bit, now in 65816 mode
        JMP   $808009   ;Example 65816 instruction
```

24.15 CAUTION #15

When utilizing the high speed mode (3.58MHz), perform a dummy jump at the start of every vector to change the Program Bank Register to the upper banks ($80 or above). Refer to the following program example.

Example:

```
                ORG     $808000
RESET
                SEI
                CLC
                XCE
                JMP     ~RESETFAST      ;Dummy jump to change PBR
RESETFAST                               ;RESETFAST belongs to bank $80
                ...
                ...
NMI
                JMP     ~NMIFAST
NMIFAST         ...
                ...
```

24.16 CAUTION #16

When restarting controller read after it has been temporarily disabled, the user program should confirm that the buttons have been released before accepting the button inputs.

Some licensed controllers latch the last data which was received after disabling controller read. This data is held for about 3 fields (50 msec) into the next controller read sequence. This performance as compared to Nintendo's standard controller performance is demonstrated in the table below.

User Operation	No Operation	"B" Button	No Operation	"A" Button
Nintendo Controller Output		B B	N/A	A A A A
Output of Some Licensed Controllers		B B	N/A	B B B A
Controller Read	Ensble		Disable	Enable

Note: 1 field (16.6 msec)

For instance, if the software is programmed as follows;

1. Enter the room when "B" button is pressed.
2. Disable controller read while changing screen data.
3. The room appears and enable controller read.
4. Exit the room when "B" button is pressed.

the player will immediately exit the room.

This problem can be resolved in 2 different ways, as described below.

24.16.1 EDGE DETECTION

If "edge detection" is used for processing controller data instead of "level detection", the above problem can be avoided. The following sample program illustrates edge detection. The difference between controller (Cont) and trigger (Trig) data in the sample program is shown in the table below.

Cont	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0
Trig	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Note: 0 = Off
1 = On

(SAMPLE PROGRAM)

```
;-------------------------------------------------------------------------------------------------------;
;- RAM Definition
;-------------------------------------------------------------------------------------------------------;
Cont1L          ds      1               ; Controller #1 data low byte
Cont1H          ds      1               ; Controller #1 data high byte
Cont2L          ds      1               ; Controller #2 data low byte
Cont2H          ds      1               ; Controller #2 data high byte
Trig1L          ds      1               ; Trigger data of controller #1
Trig1H          ds      1               ;
Trig2L          ds      1               ;Trigger data of controller #2
Trig2H          ds      1               ;
;-------------------------------------------------------------------------------------------------------;
;- Read Controller
;-------------------------------------------------------------------------------------------------------;
RdCont;
                push
                a8                      ; Accumulator 8-bit
RdCont_Wait1
                LDA     HVBJoy          ; <4212>
                AND     #%00000001      ; Wait JOY-C Enable : D0=0
                BEQ     RdCont_Wait1
RdCont_Wait2
                LDA     HVBJoy
                AND     #%00000001
                BNE     RdCont_Wait2
                a16                     ; Accumulator 16-bit
                i16                     ; Index 16-bit
RdCont_Cont1
                LDY     Cont1L          ; Keep last data in "IY"
                LDA     Joy1L           ; <4218> (Cont1-L)
                STA     Cont1L          ; Store new controller data
                TYA                     ; <edge detection>
                EOR     Cont1L
                AND     Cont1L
                STA     Trig1L          ; Store trigger data
RdCont_Cont2
                LDY     Cont2L          ; Keep last data in "IY"
                LDA     Joy2L           ; <421AH> (Cont2-L)
                STA     Cont2L          ; Store new controller data
                TYA                     ; <edge detection>
                EOR     Cont2L
                AND     Cont2L
                STA     Trig2L          ; Store trigger data
                pop
                RTS
```

24.16.2 ALTERNATE METHOD

The problem may be avoided by ignoring controller data for about 3 fields, after restarting controller read. Since programming becomes very complicated, increasing the risk of program bugs, this method is not recommended.

If controller read is disabled for 1~2 fields, the consumer cannot press a button quickly enough to cause a problem. This configuration is illustrated in the table below.

User Operation	No Operation	"B" Button	No Operation	"A" Button
Nintendo Controller Output		B B N/A B B		A A A A
Output of Some Licensed Controllers		B B N/A B B		A A A A
Controller Read	Ensble	Disable	Enable	

Note: 1 field (16.6 msec)

THIS PAGE INTENTIONALLY
LEFT BLANK

Chapter 25. Documented Problems

The following paragraphs describe system problems which have been identified and provides solutions for the problems listed.

25.1 PROBLEM 1

25.1.1 SYMPTOM

If H-DMA starts at about the same time that General Purpose DMA finishes, sometimes the CPU will cease to operate properly or H-DMA will not be correctly implemented (S-CPU ver. 1).

This could happen if General Purpose DMA finishes during the first 2.24 µs of the H - Blank period on lines 0 - 224 (239), while H-DMA is being used. It can also happen at the beginning of line 0, as well. ❇

❇ The real time trace function of the ICE can be utilized to confirm the timing.

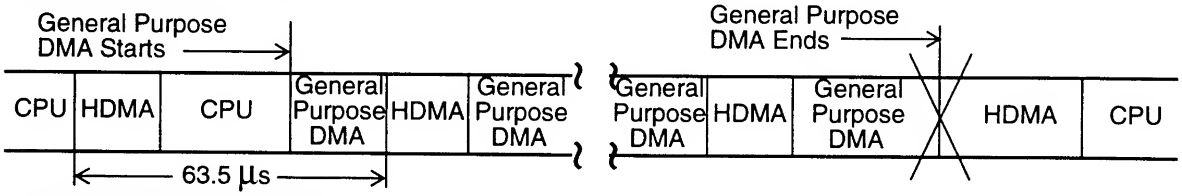
25.1.2 SOLUTION

This problem will not happen if General Purpose DMA is used only during V - Blank or if H-DMA starts in the middle of data transfer of General Purpose DMA. It does not happen if H-DMA is not being used.

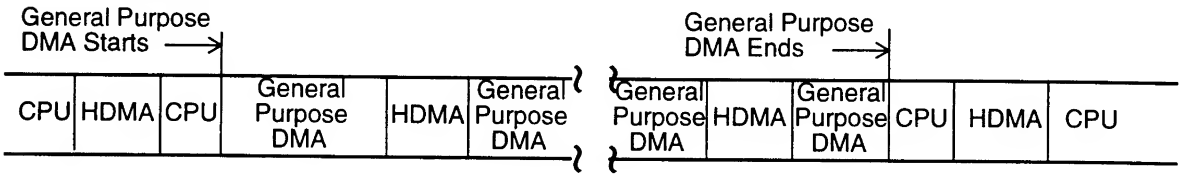
This problem can also be avoided by adjusting the time at which General Purpose DMA begins and/or decreasing the number of bytes transferred. The H and V count timers can be utilized to determine the start time of General Purpose DMA. One line takes 63.5 µs and the value of one count on the H count timer is equivalent to 0.186 µs.

The end timing of the General Purpose DMA changes depending on the amount of transferred data of H-DMA which happens in the middle of data transfer of the General Purpose DMA.

When the problem occurs:



When avoiding the problem:



25.2 PROBLEM 2

25.2.1 SYMPTOM

When the size of OBJ 0 is 16 x 16, 32 x 32, or 64 x 64, and its horizontal position is 0 through 255, and there are other objects present with negative horizontal positions (they are not displayed on the screen), the Time Over Flag will become 1 (S-PPU1 ver. 1).

25.2.2 SOLUTION

The cause is being examined.

(NCL PG 43b)

Chapter 26. Register Clear (Initial Settings)

(This is a recommended setting for beginners. It is not necessary to perform register clear exactly this way. However, the register status is not stable when power is turned on and initial settings must be performed).

ADDRESS (HEX)	DATA (HEX)
<2100>	8 F (Forced Blank)
<2101>	0 0
<2102>	0 0
<2103>	0 0
<2104>	(OAM Data)
<2105>	0 0
<2106>	0 0
<2107>	0 0
<2108>	0 0
<2109>	0 0
<210A>	0 0
<210B>	0 0
<210C>	0 0
	(Low) (High)
<210D>	0 0 0 0
<210E>	0 0 0 0
<210F>	0 0 0 0
<2110>	0 0 0 0
<2111>	0 0 0 0
<2112>	0 0 0 0
<2113>	0 0 0 0
<2114>	0 0 0 0
<2115>	8 0
<2116>	0 0
<2117>	0 0
<2118>	(VRAM Data)
<2119>	(VRAM Data)
<211A>	0 0
<211B>	0 0 0 1
<211C>	0 0 0 0
<211D>	0 0 0 0
<211E>	0 0 0 1
<211F>	0 0 0 0

ADDRESS (HEX)	DATA (HEX)
<2120>	0 0 0 0
<2121>	0 0
<2122>	(CG Data)
<2123>	0 0
<2124>	0 0
<2125>	0 0
<2126>	0 0
<2127>	0 0
<2128>	0 0
<2129>	0 0
<212A>	0 0
<212B>	0 0
<212C>	0 0
<212D>	0 0
<212E>	0 0
<2130>	3 0
<2131>	0 0
<2132>	E 0
<2133>	0 0
<4200>	0 0
<4201>	F F
<4202>	0 0
<4203>	0 0
<4204>	0 0
<4205>	0 0
<4206>	0 0
<4207>	0 0
<4208>	0 0
<4209>	0 0
<420A>	0 0
<420B>	0 0
<420C>	0 0
<420D>	0 0

(NCL PG 44)

Chapter 27. PPU Registers

ADDRESS: 2100H
NAME: INIDISP
CONTENTS: INITIAL SETTINGS FOR SCREEN

D7	D6	D5	D4	D3	D2	D1	D0
BLANK-ING				FADE IN/OUT (0~15)			
				F3	F2	F1	FO

2100H

SCREEN BRIGHTNESS: Determine the screen brightness (16-stages)

F3	F2	F1	F0	BRIGHTNESS
1	1	1	1	BRIGHT
1	1	1	0	↓
		≀		
0	0	0	0	DARK

BLANKING: FORCED BLANKING — 0: NON-BLANKING, 1: BLANKING

ADDRESS: 2101H
NAME: OBJSEL
CONTENTS: OBJECT SIZE & OBJECT DATA AREA DESIGNATION

D7	D6	D5	D4	D3	D2	D1	D0
OBJ	SIZE SELECT		OBJ NAME SELECT		OBJ NAME BASE ADDR		
S2	S1	S0	N1	N0	BA-2	BA-1	BA-0

2101H

For Expansion Norm = 0

OBJECT BASE ADDRESS (UPPER 3 BIT)
Designate the segment (8K-word/ segment) address which the OBJ data is stored in the VRAM. (See pages A-1 and A-2)

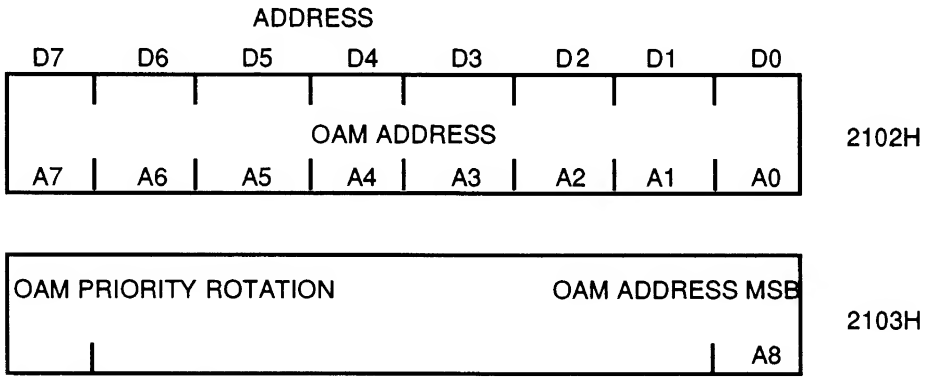
OBJECT DATA AREA SELECT
The upper 4K-word out of the area (8K-word) designated by "Object Base Address" is assigned as the Base Area, and the area of the lower 4K-word combined with its Base Area can be selected. (See pages A-1 and A-2)

OBJECT SIZE: DESIGNATE OBJECT SIZE (See pages A-3 and A-4)

S2	S1	S0	OBJ SIZE 0 (SM)	OBJ SIZE 1 (LG)
0	0	0	8 DOT	16 DOT
0	0	1	8 DOT	32 DOT
0	1	0	8 DOT	64 DOT
0	1	1	16 DOT	32 DOT
1	0	0	16 DOT	64 DOT
1	0	1	32 DOT	64 DOT

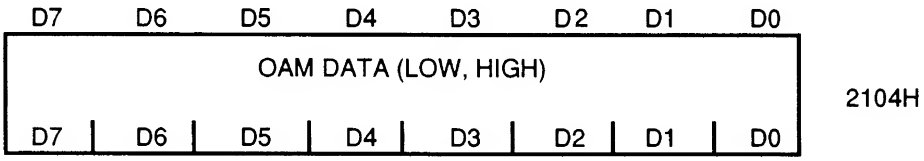
(NCL PG 46)

ADDRESS: 2102H /2103H
NAME: OAMADDL / OAMADDH
CONTENTS: ADDRESS FOR ACCESSING OAM (OBJECT ATTRIBUTE MEMORY)



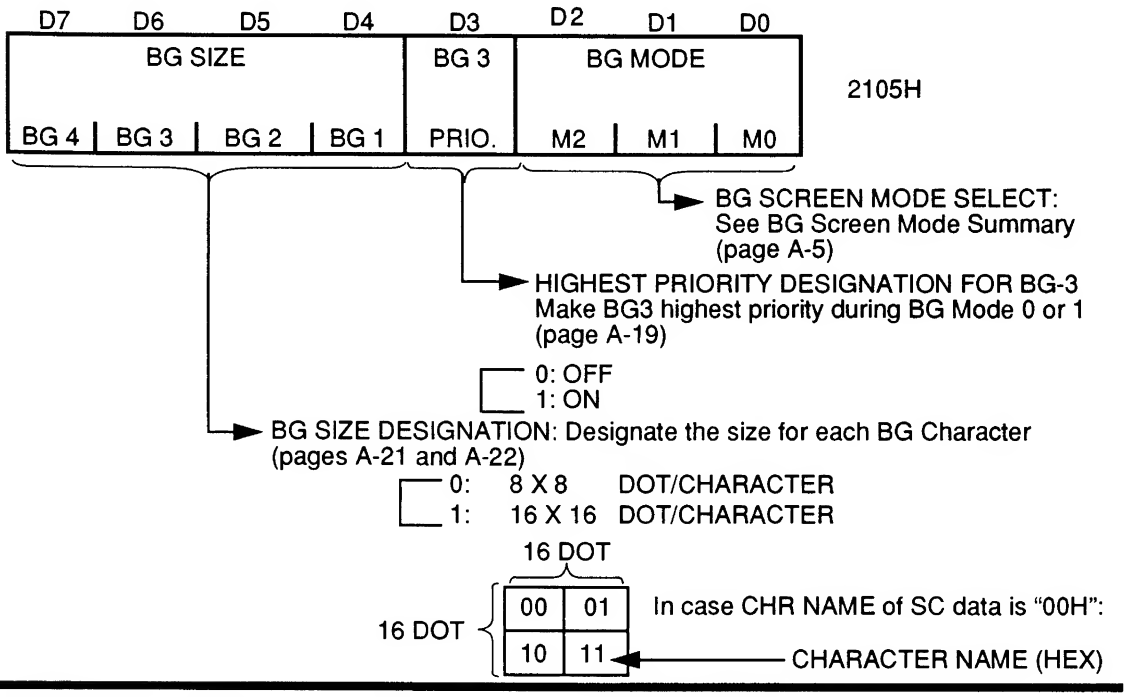
- This is the INITIAL ADDRESS to be set in advance when reading from or writing to the OAM.
- To set the OBJ priority order, write "1" to D7 (OAM Priority Rotation) of register <2103H> and set the highest priority OBJ number (0 ~ 127) to D1 ~ D7 of register <2102H> (refer to "Priority Order Shifting").
- The address which has been set just before every field (beginning with V-BLANK) will be set again to registers <2102H> <2103H> automatically. However, the address cannot be set automatically during Forced Blank period.

ADDRESS: 2104H
NAME: OAM DATA
CONTENTS: DATA FOR OAM WRITE

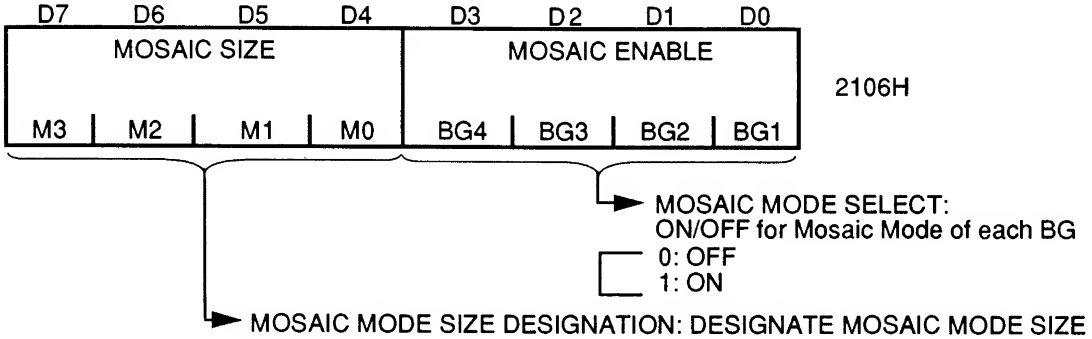


- This is the OAM data to be written to any address of the OAM (refer to page A-3).
- After register <2102H> or <2103H> is accessed, the data must be written in the order of Lower 8-bit and Upper 8-bit of register <2104H>. The OAM address will be increased automatically when the OAM data is written in the order of LOW to HIGH.
- The data can be written only during a V-BLANK or FORCED BLANK period.

ADDRESS: 2105H
NAME: BG MODE
CONTENTS: BG MODE & CHARACTER SIZE SETTINGS



ADDRESS: 2106H
NAME: MOSAIC
CONTENTS: SIZE & SCREEN DESIGNATION FOR MOSAIC DISPLAY



256 MODE

M3	M2	M1	M0	SIZE
0	0	0	0	1 X 1 DOT
0	0	0	1	2 X 2 DOT
0	0	1	0	3 X 3 DOT
		•		•
		•		•
		•		•
1	1	1	0	15 X 15 DOT
1	1	1	1	16 X 16 DOT

512 MODE (H X V)

M3	M2	M1	M0	SIZE ( ): NON-INTERLACE
0	0	0	0	2 X 2 DOT (2 X 1 DOT)
0	0	0	1	4 X 4 DOT (4 X 2 DOT)
0	0	1	0	6 X 6 DOT (6 X 3 DOT)
		•		•
		•		•
		•		•
1	1	1	0	30 X 30 DOT (30 X 15 DOT)
1	1	1	1	32 X 32 DOT (32 X 16 DOT)

(NCL PG 48)

ADDRESS: 2107H / 2108H / 2109H / 210AH
NAME: BG1SC / BG2SC / BG3SC / BG4SC
CONTENTS: ADDRESS FOR STORING SC-DATA OF EACH BG & SC SIZE DESIGNATION (MODE 0 ~ 6)

D7	D6	D5	D4	D3	D2	D1	D0	
BG1 SC BASE ADDRESS						BG1 SC SIZE		2107 H
A5	A4	A3	A2	A1	A0	S1	S0	
BG2 SC BASE ADDRESS						BG2 SC SIZE		2108 H
A5	A4	A3	A2	A1	A0	S1	S0	
BG3 SC BASE ADDRESS						BG3 SC SIZE		2109H
A5	A4	A3	A2	A1	A0	S1	S0	
BG4 SC BASE ADDRESS						BG4 SC SIZE		210A H
A5	A4	A3	A2	A1	A0	S1	S0	

→ SCREEN SIZE DESIGNATION (pages A-21 and A-22) Designate Background Screen Size

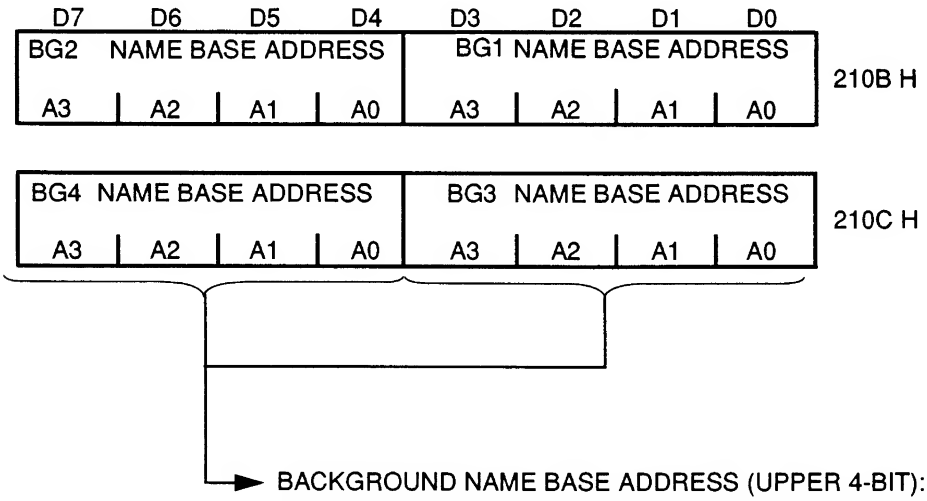
→ BACKGROUND SCREEN BASE ADDRESS (UPPER 6-BIT) Designate the segment in which BG-SC data in the VRAM is stored. (1K-WORD/SEGMENT)

SCREEN SIZE & SCREEN REPETITION

S1	S0	SCREEN SIZE
0	0	SC0 SC0 / SC0 SC0
0	1	SC0 SC1 / SC0 SC1
1	0	SC0 SC0 / SC1 SC1
1	1	SC0 SC1 SC0 / SC2 SC3 SC2 / SC0 SC1 SC0

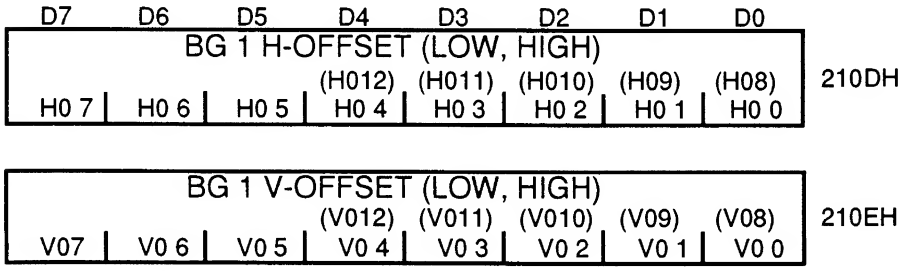
(NCL PG 49)

ADDRESS: 210BH / 210CH
NAME: BG12NBA / BG34NBA
CONTENTS: BG CHARACTER DATA AREA DESIGNATION



BACKGROUND NAME BASE ADDRESS (UPPER 4-BIT):
Designate the segment address in the VRAM in which BG character data is stored. (4K-WORD/SEGMENT)

ADDRESS: 210DH / 210EH
NAME: BG1H0FS / BG1V0FS
CONTENTS: H/V SCROLL VALUE DESIGNATION FOR BG-1



- 10-Bit maximum (0 ~ 1023) can be designated for H/V scroll value. (The size of 13-Bit maximum {-4096 ~ 4095} can be designated in MODE -7).
  (pages A-10 and A-11)

- By writing to the register twice, the data can be set in the order of Low and High.

ADDRESS: 210FH / 2110 H / 2111 H / 2112 H / 2113 H / 2114 H
NAME: BG2H0FS / BG2V0FS / BG3H0FS / BG3V0FS / BG4H0FS / BG4V0FS
CONTENTS: H/V SCROLL VALUE DESIGNATION FOR BG-2, 3, 4

D7	D6	D5	D4	D3	D2	D1	D0	
BG H-OFFSET (LOW, HIGH)						(H0 9)	(H0 8)	210FH 2111H 2113H
H0 7	H0 6	H0 5	H0 4	H0 3	H0 2	H0 1	H0 0	

D7	D6	D5	D4	D3	D2	D1	D0	
BG V-OFFSET (LOW, HIGH)						(V0 9)	(V0 8)	2110H 2112H 2114H
V07	V0 6	V0 5	V0 4	V0 3	V0 2	V0 1	V0 0	

- 10-Bit maximum (0 ~ 1023) of the H/V scroll value can be designated (page A-10)
- By writing to the register twice, the data can be set in the order of Low- and High.

ADDRESS: 2115H
NAME: VMAINC
CONTENTS: VRAM ADDRESS INCREMENT VALUE DESIGNATION

D7	D6	D5	D4	D3	D2	D1	D0	
H/L INC				V-RAM ADDRESS FULL GRAPHIC G1	G0	SEQUENCE MODE SC INCREMENT I1	I0	2115H

Designate the increment value for the VRAM address. (page A-8)

G1	G0	I1	I0	INCREMENT VALUE
0	1	0	0	Increment by 8 (for 32 times) (2-Bit Formation)
1	0	0	0	Increment by 8 (for 64 times) (4-Bit Formation)
1	1	0	0	Increment by 8 (for 128 times) (8-Bit Formation)
0	0	0	0	Address Increments 1 BY 1
0	0	0	1	Address Increments 32 BY 32
0	0	1	0	Address Increments 128 BY 128
0	0	1	1	Address Increments 128 BY 128

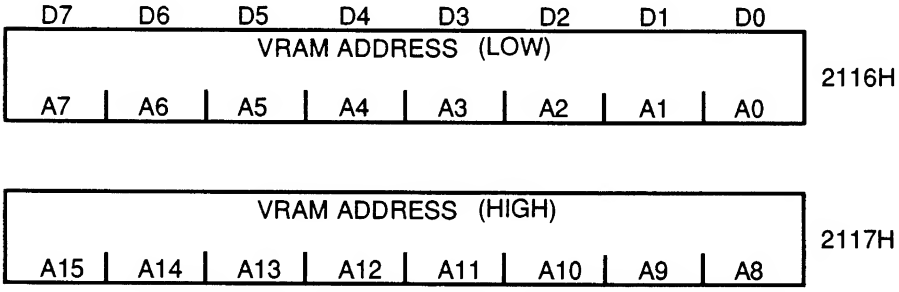
Designate the increment timing for the address

0: The address will be increased after the data has been written to register <2118H> or the data has been read from register <2139>.

1: The address will be increased after the data has been written to register <2119H> or the data has been read from register <213AH>.

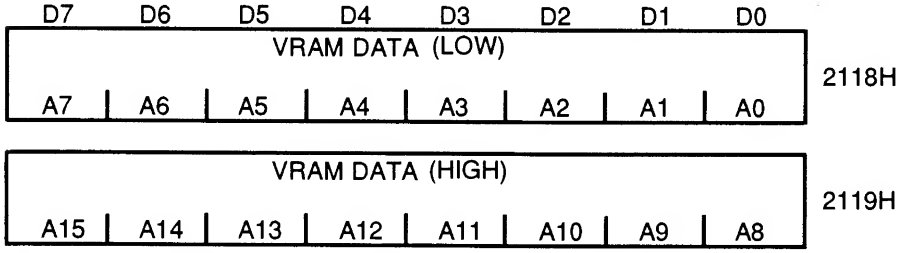
(NCL PG 51)

ADDRESS: 2116H / 2117H
NAME: VMADDL / VMADDH
CONTENTS: ADDRESS FOR VRAM READ AND WRITE



- This is the initial address for reading from the VRAM or writing to the VRAM.
- The data is read or written by the address set initially, and every time the data is read or written, the address will be increased automatically.
- The value to be increased is determined by "SC INCREMENT" of register <2115H> and the setting value of the "FULL GRAPHIC."

ADDRESS: 2118H / 2119H
NAME: VMDATAL / VMDATAH
CONTENTS: DATA FOR VRAM WRITE



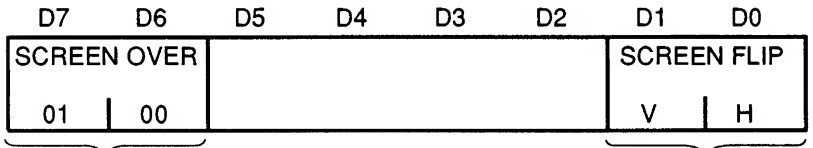
- This is the screen data and character data (BG & OBJ), which can be written to any address in the VRAM.
- According to the settings of register <2115H> "H/L INC," the data can be written to the VRAM as follows:

H/L INC	WRITE TO REGISTER	OPERATION
0	Write to <2118H> only	The data is written to lower 8-bit of the VRAM and the address will be increased automatically.
1	Write to <2119H> only	The data is written to upper 8-bit of the VRAM and the address will be increased automatically.
0	Write in the order of <2119H> and <2118H>	When the data is set in the order of upper and lower, the address will be increased.
1	Write in the order of <2118H> and <2119H>	When the data is set in the order of lower and upper, the address will be increased

NOTE: The data can be written only during V-BLANK or FORCED BLANK period.

(NCL PG 52)

ADDRESS: 211AH
NAME: M7SEL
CONTENTS: INITIAL SETTING IN SCREEN MODE-7



HORIZONTAL/VERTICAL FLIP: H-FLIP/V-FLIP in the Screen Mode-7

V	H	DISPLAY
0	0	Normal display
0	1	H-Directional Flip only
1	0	V-Directional Flip only
1	1	Both H & V Directional Flip

The following process is made if the screen to be displayed is outside of the screen area.

01	00	PROCESS OUT OF AREA
0	0	Screen repetition if outside of screen area
1	0	Outside of the screen area is the Back Drop Screen in single color
1	1	Character #0 repetition if outside of screen area

(NCL PG 53)

ADDRESS: 211B H / 211C H / 211D H / 211E H / 211F H / 2120 H
NAME: M7A / M7B / M7C / M7D / M7X / M7Y
CONTENTS: ROTATION/ENLARGEMENT/REDUCTION IN MODE-7, CENTER COORDINATE SETTINGS & MULTIPLICAND/MULTIPLIER SETTINGS OF COMPLEMENTARY MULTIPLICATION

D7	D6	D5	D4	D3	D2	D1	D0	
MATRIX PARAMETER A (LOW, HIGH)								
(MP15) MP 7	(MP14) MP 6	(MP13) MP 5	(MP12) MP 4	(MP11) MP 3	(MP10) MP 2	(MP9) MP 1	(MP8) MP 0	211 BH
MATRIX PARAMETER B (LOW, HIGH)								
(MP15) MP 7	(MP14) MP 6	(MP13) MP 5	(MP12) MP 4	(MP11) MP 3	(MP10) MP 2	(MP9) MP 1	(MP8) MP 0	211 CH
MATRIX PARAMETER C (LOW, HIGH)								
(MP15) MP 7	(MP14) MP 6	(MP13) MP 5	(MP12) MP 4	(MP11) MP 3	(MP10) MP 2	(MP9) MP 1	(MP8) MP 0	211 DH
MATRIX PARAMETER D (LOW, HIGH)								
(MP15) MP 7	(MP14) MP 6	(MP13) MP 5	(MP12) MP 4	(MP11) MP 3	(MP10) MP 2	(MP9) MP 1	(MP8) MP 0	211 EH

- The 8-bit data should be written twice in the order of lower and upper. Then, the parameter of rotation, enlargement and reduction should be set by its 16-bit data.
- The value down to a decimal point should be set to the lower 8-bit. The most significant bit of the upper 8-bit is for the signed bit. (MP15 is the signed bit. There is a decimal point between M7 & M8.)
- FORMULA FOR ROTATION/ENLARGEMENT/REDUCTION (Refer to Rotation/Enlargement/Reduction in Appendix A.).

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_1 - X_0 \\ Y_1 - Y_0 \end{bmatrix} + \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

$A = \cos\gamma x (1 / \alpha)$, $B = \sin\gamma x (1 / \alpha)$, $C = -\sin\gamma x (1 / \beta)$, $D = \cos\gamma x (1 / \beta)$

γ: Rotation angle α: Reduction Rates for X (H) β: Reduction Rates for Y (v)

$X_0 \bullet Y_0$: Center Coordinate, $X_1 \bullet Y_1$: Display Coordinate,

$X_2 \bullet Y_2$: Coordinate Before Calculation

- Set the value of "A" to the register <211BH>. In the same way, set "B~D" to the register <211CH> ~ <211EH>.
- The complementary multiplication (16-bit x 8-bit) can be done by using registers <211BH> <211CH>. When setting 16-bit data to register <211BH> (must be written twice) and 8-bit data to register <211CH> (must be written only once), the multiplication result can be indicated rapidly by reading registers <2134H> ~ <2136H>.

D7	D6	D5	D4	D3	D2	D1	D 0	
CENTER POSITION X_0 (LOW, HIGH)								211 FH
			(X12)	(X11)	(X10)	(X9)	(X8)	
X 7	X 6	X 5	X 4	X 3	X 2	X 1	X 0	
CENTER POSITION Y_0 (LOW, HIGH)								212 0H
			(Y12)	(Y11)	(Y10)	(Y9)	(Y8)	
Y 7	Y6	Y 5	Y 4	Y 3	Y 2	Y 1	Y 0	

- The center coordinate (X_0 Y_0) for Rotation/Enlargement/Reduction can be designated by this register.
- The coordinate value of X_0 & Y_0 can be designated by 13-bit (complement of 2).
- This register requires that the lower 8-bit set first and the upper 5-bit is set. Therefore, 13-bit data in total can be set.

ADDRESS: 2121H
NAME: CGADD
CONTENTS: ADDRESS FOR CG-RAM READ AND WRITE

D7	D6	D5	D4	D3	D2	D1	D0	
CG RAM ADDRESS								2121H
A7	A6	A5	A4	A3	A2	A1	A0	

- This is the initial address for reading from the CG-RAM or writing to the CG-RAM.
- The data is read by address set initially, and every time the data is read or written, the address will be increased automatically.

ADDRESS: 2122H
NAME: CGDATA
CONTENTS: DATA FOR CG-RAM WRITE

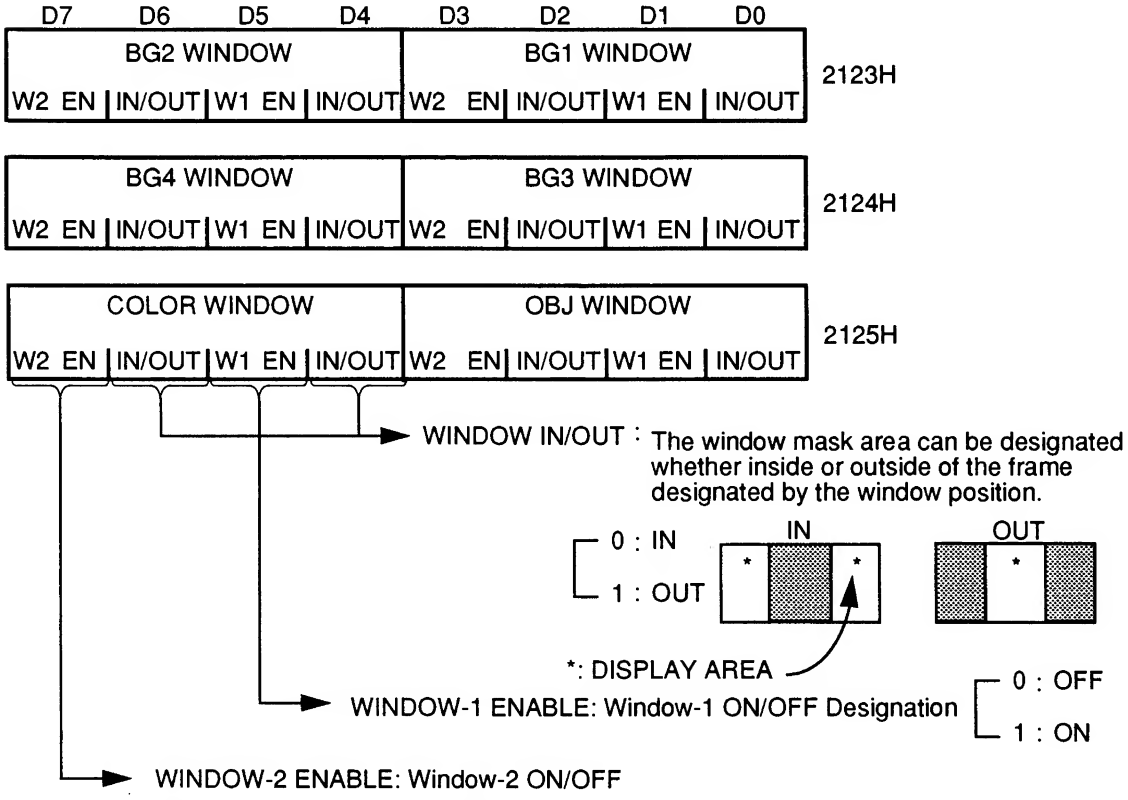
D7	D6	D5	D4	D3	D2	D1	D0	
CG RAM DATA (LOW, HIGH)								2122H
D7	(D14) D6	(D13) D5	(D12) D4	(D11) D3	(D10) D2	(D9) D1	(D8) D0	

- This is the color generator data to be written at any address of the CG-RAM.
- The mapping of BG1 ~ BG 4 and OBJ data in the CG-RAM will be determined, which is performed by every mode selected by "BG MODE" of register <2105H>. (See page A-17)
- There are the color data of 8 palettes for each screen of BG1 ~ BG4. The palette selection is determined by 3-bit of the SC data "COLOR."(Refer to page A-10)
- Because the CG-RAM data is 15-bit/word, it is necessary to set lower 8-bit first to this register and then upper 7-bit should be set. When both lower and upper are set, the address will be increased by 1 automatically.

  NOTE: After the address is set, the data should be written in the order of low, then high. This is similar to the OAM Data register.

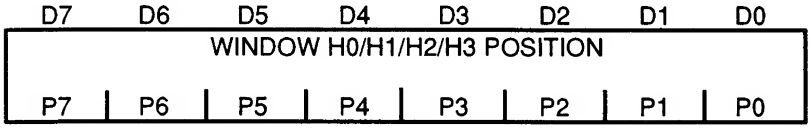
  NOTE: The data can be written only during H/V BLANK or FORCED BLANK period.

ADDRESS: 2123H / 2124H / 2125H
NAME: W12SEL/ W34SEL/ WOBJSEL
CONTENTS: WINDOW MASK SETTINGS (BG1~BG4, OBJ, COLOR)



The COLOR WINDOW is a window for main and sub screen. (It is related to the register <2130H>).

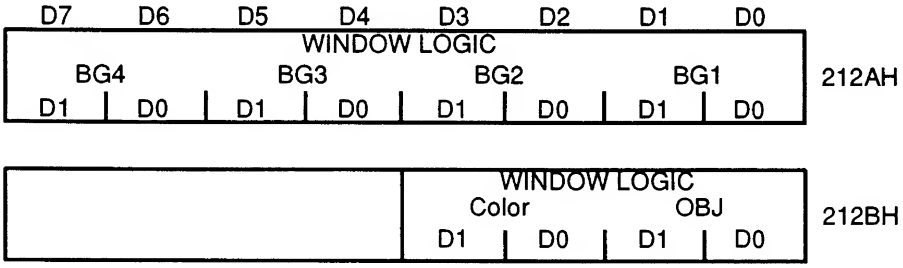
ADDRESS: 2126H/ 2127H/ 2128H/ 2129H
NAME: WH0/ WH1/ WH2/ WH3
CONTENTS: WINDOW POSITION DESIGNATION (Refer to page A-18)



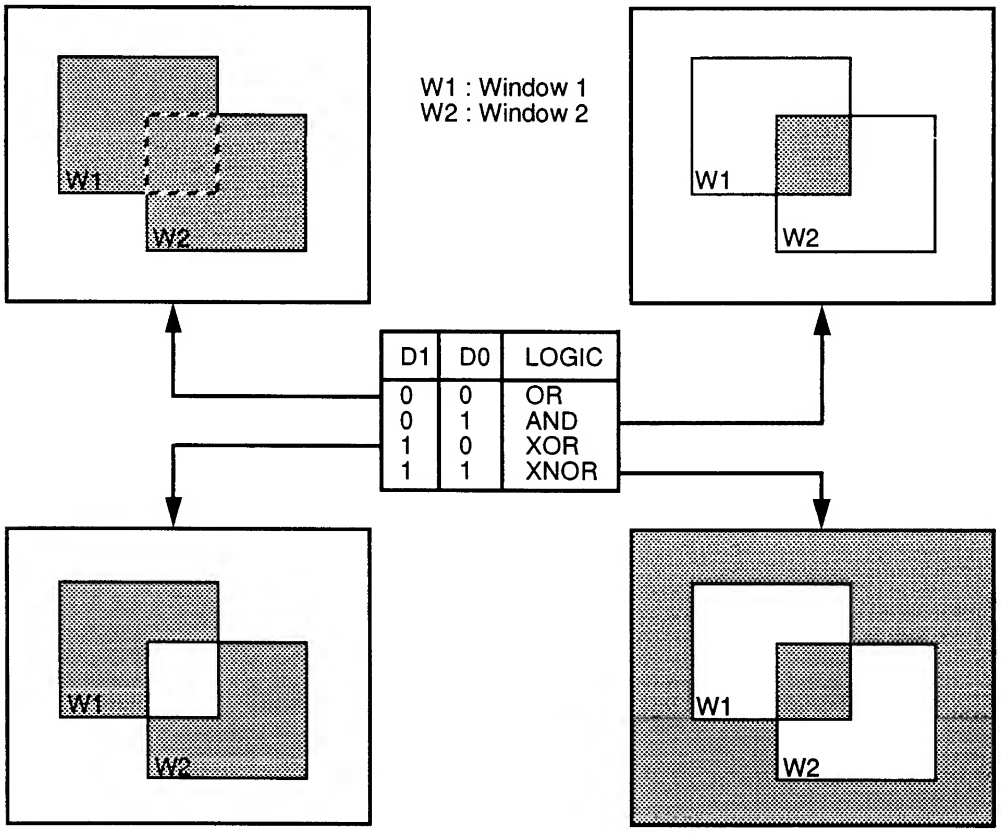
WINDOW H0 POSITION <2126H>:WINDOW-1Left Position Designation. It can be set in range 0 ~255
WINDOW H1 POSITION <2127H>:WINDOW-1 Right Position Designation. It can be set in range 0 ~255
WINDOW H2 POSITION <2128H>:WINDOW-2 Left Position Designation. It can be set in range 0 ~255
WINDOW H3 POSITION <2129H>:WINDOW-2 Right Position Designation. It can be set in range 0 ~255

NOTE: If 'LEFT POSITION SETTING VALUE> RIGHT POSITION VALUE" is assumed, there will be no range of the window.

ADDRESS: 212AH/ 212BH
NAME: WBGLOG/ WOBJLOG
CONTENTS: MASK LOGIC SETTINGS FOR WINDOW-1 & 2 ON EACH SCREEN



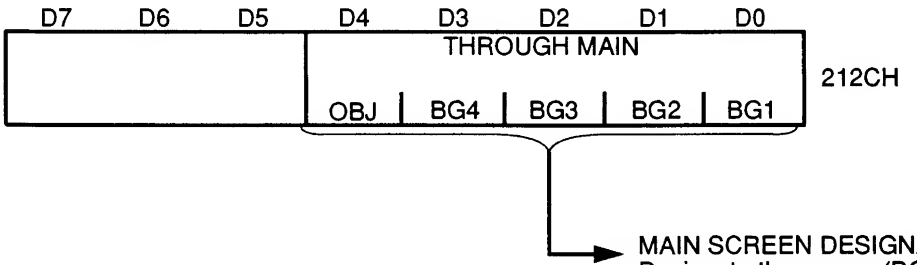
WINDOW LOGIC: SET MASK LOGIC FOR WINDOW-1 & 2
When both Window-1 and Window-2 are "IN," the shaded portion will be masked as follows:



D1	D0	LOGIC
0	0	OR
0	1	AND
1	0	XOR
1	1	XNOR

NOTE: "IN/OUT" of registers <2123H> <2124H> <2125H> becomes the "NOT logic" for each Window-1 and Window-2.

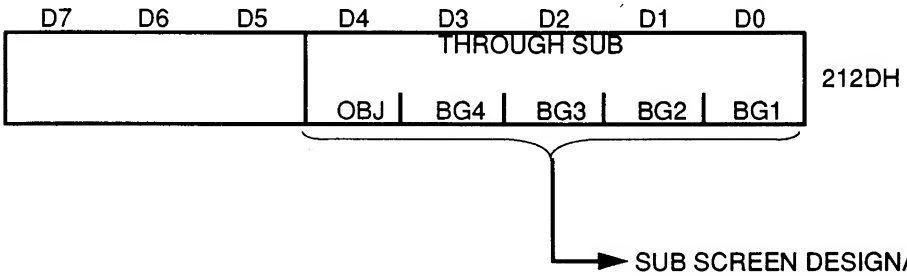
ADDRESS: 212CH
NAME: TM
CONTENTS: MAIN SCREEN DESIGNATION



MAIN SCREEN DESIGNATION:
Designate the screen (BG1 ~ BG4, OBJ) to be displayed as the Main Screen.
Designate the screen to be added for the screen addition/subtraction .

0 : DISABLE
1 : ENABLE

ADDRESS: 212DH
NAME: TS
CONTENTS: SUB SCREEN DESIGNATION



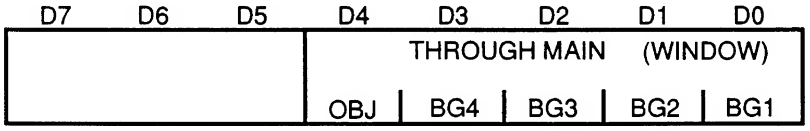
SUB SCREEN DESIGNATION:
Designate the screen (BG1 ~ BG4, OBJ) to be displayed as SUB-Screen.
Designate the screen to be added for the screen addition/subtraction

0 : DISABLE
1 : ENABLE

NOTE: When the screen addition/subtraction is functioning, the SUB screen is a screen to be added or subtracted against the MAIN screen.

(NCL PG 58)

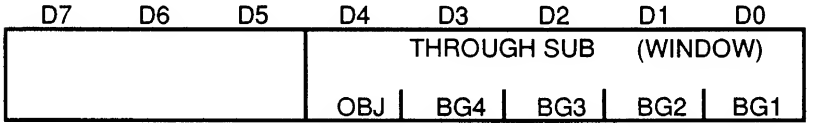
ADDRESS: 212EH
NAME: TMW
CONTENTS: WINDOW MASK DESIGNATION FOR MAIN SCREEN



WINDOW MASK DESIGNATION FOR MAIN SCREEN:
In the window area designated by register <2123H> ~ <2129H>, the screen to be displayed can be designated, which is selected among the Main screen designated by register <212CH>.

0 : DISABLE
1 : ENABLE

ADDRESS: 212FH
NAME: TSW
CONTENTS: WINDOW MASK DESIGNATION FOR SUB SCREEN



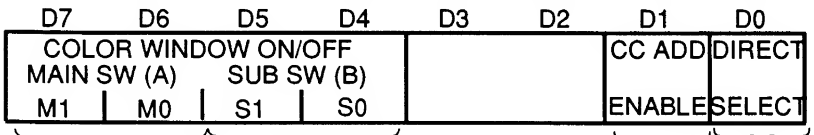
WINDOW MASK DESIGNATION FOR SUB SCREEN:
In the window area designated by register <2123H> ~ <2129H>, the screen to be displayed can be designated, which is selected among the Sub screen designated by register <212DH>.

0 : DISABLE
1 : ENABLE

NOTE: When the screen addition/subtraction is functioning, the SUB screen is a screen to be added or subtracted against the MAIN screen.

(NCL PG 59)

ADDRESS: 2130H
NAME: CGSWSEL
CONTENTS: INITIAL SETTINGS FOR FIXED COLOR ADDITION OR SCREEN ADDITION



DIRECT SELECT (Refer to p. A-17):
The VRAM data (Color & Character data) become the color data directly. (Only when mode-3, 4 & 7 are selected.)

0 : DISABLE
1 : ENABLE

FIXED COLOR ADDITION/SUBTRACTION ENABLE:
Designate whether 2 kinds of the data should be added/subtracted from each other or not, which are the fixed color set by register <2132H> and the color data which is set to the CGRAM.

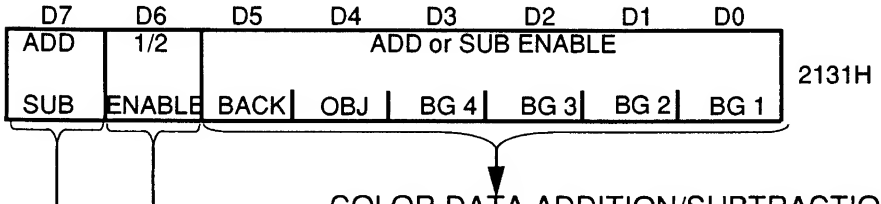
0 : ADDITION/SUBTRACTION FOR FIXED COLOR
1 : ADDITION/SUBTRACTION FOR SUB SCREEN

COLOR WINDOW ON/OFF MAIN/SUB SWITCH:
When the Color Window is functioning, the assignment of the window area for MAIN and SUB screens can be designated.

M1 (S1)	M0 (S0)	FUNCTION
0	0	ON (All the time)
0	1	ON (Inside window only)
1	0	ON (Outside window only)
1	1	OFF (All the time)

(NCL PG 60)

ADDRESS: 2131H
NAME: CGADSUB
CONTENTS: ADDITION/SUBTRACTION & SUBTRACTION DESIGNATION FOR EACH BG SCREEN OBJ & BACKGROUND COLOR



COLOR DATA ADDITION/SUBTRACTION ENABLE:

Designate the color data of BG1 ~ BG4, OBJ, or Back in the main screen for addition/subtraction of the Sub screen color data (or fixed color data.)

0:DISABLE
1:ENABLE (Addition/Subtraction function: ON)

Note: When OBJ is designated, the Addition/Subtraction function is available only when the OBJ color palette is 4 through 7.

"1/2 OF COLOR DATA" DESIGNATION:

When color constant addition/subtraction or screen addition/subtraction is performed, designate whether the RBG result in the addition/subtraction area should be "1/2" or not. The back (color constant) area on the Sub·screen, will not become "1/2"

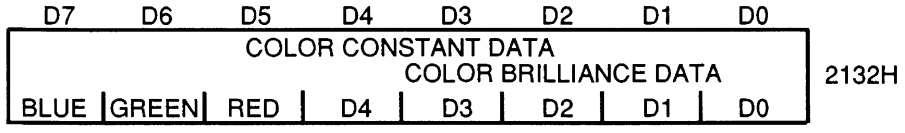
0 : DISABLE
1 : ENABLE (1/2 function: ON)

COLOR DATA ADDITION/SUBTRACTION SELECT:

In the case of executing screen addition/subtraction, designate either addition or subtraction mode.

0 : ADDITION MODE SELECT
1 : SUBTRACTION MODE SELECT

ADDRESS: 2132H
NAME: COLDATA
CONTENTS: FIXED COLOR DATA FOR FIXED COLOR ADDITION/SUBTRACTION



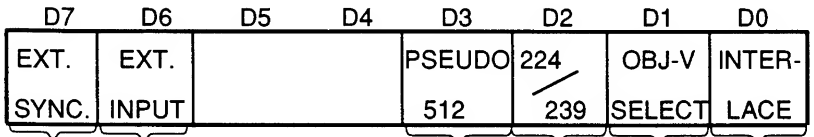
COLOR CONSTANT DATA:
Set the color constant data for color constant addition/subtraction.

COLOR DESIGNATION: Bit for Selecting Desired Color

R/G/B brightness should be set using 5-bit data. Example:

RED : C0H, 3FH (B=00H, G=00H, R=1FH)
GREEN : A0H, 5FH (B=00H, G=1FH, R=00H)
BLUE : 60H, 9FH (B=1FH, G=00H, R=00H)
WHITE : FFH
BLACK : E0H

ADDRESS: 2133H
NAME: SETINI
CONTENTS: SCREEN INITIAL SETTING



SCANNING INTERLACE(1) NON/INTERLACE(0) SELECTION:
(Related to <2105H>)

OBJ V-DIRECTION DISPLAY:✳
In the interlace mode, select either 1 dot per line or 1 dot repeated every 2 lines. If "1" is written, the OBJ appears to be reduced to half its vertical size.

BG V-DIRECTION DISPLAY:
Switch the display line of a field to 224-Line or 239-Line. (In case of interlace mode, it will be doubled dot.)
0 : 224 LINES
1 : 239 LINES

HORIZONTAL PSEUDO 512 MODE:
An imaginary resolution of 512 (Horizontal) can be created by shifting the SUB screen half dot to the left, alternately every field:
0 : DISABLE
1 : ENABLE

EXTBG MODE (SCREEN EXPAND):
Enable the data supplied from the external LSI. For the Super NES, enable when the screen with priority is used on mode-7.

EXTERNAL SYNCHRONIZATION:
Used for super-imposing images, etc. Normally, "0" should be written.

✳ If "D1" is set in non-interlace mode, even and odd numbered lines of the OBJ will be displayed alternately every field.

(NCL PG 61)

ADDRESS: 2134H / 2135H / 2136H
NAME: *MPYL / *MPYM/ *MPYH
CONTENTS: MULTIPLICATION RESULT

D7	D6	D5	D4	D3	D2	D1	D0	
M P Y (LOW)								2134H
M7	M6	M5	M4	M3	M2	M1	M0	
M P Y (MID)								2135H
M15	M14	M13	M12	M11	M10	M9	M8	
M P Y (HIGH)								2136H
M23	M22	M21	M20	M19	M18	M17	M16	

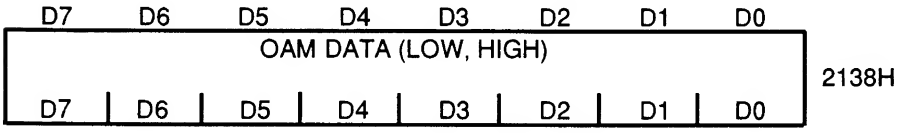
- This is a Multiplication result (complement of 2) and can be read by setting 16-bit to register <211BH> and setting 8-Bit data to register <211CH>

ADDRESS: 2137H
NAME: *SLHV
CONTENTS: SOFTWARE LATCH FOR H/V COUNTER

D7	D6	D5	D4	D3	D2	D1	D0	
SOFT LATCH FOR H/V COUNTER								2137H
SL7	SL6	SL5	SL4	SL3	SL2	SL1	SL0	

- This is a register, which generates the pulse for latching the H/V counter value.
- The H/V counter value at the point when register <2137H> is read can be latched. The data which was read is meaningless data.
- The H/V counter value latched can be referred by registers <213CH> and <213DH>.

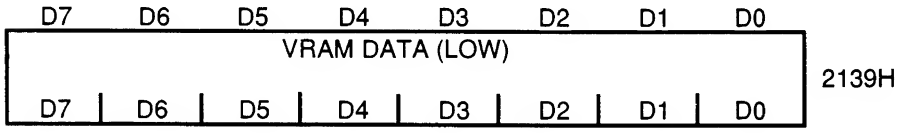
ADDRESS: 2138H
NAME: *OAMDATA
CONTENTS: READ DATA FROM OAM

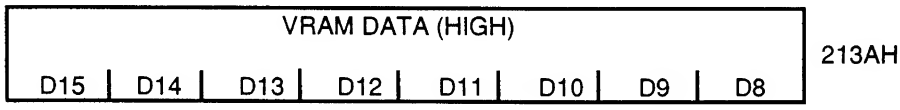


- This is a register, which can read the data at any address of the OAM.
- When the address is set to register <2102H> <2103H> and register <2138H> is also accessed, the data can be read in the order of Low 8-Bit/High 8-Bit. Afterward, the address will be increased automatically, and the data of the next address can be read.

NOTE: The data can be read only during H/V BLANK or FORCED BLANK period.

ADDRESS: 2139H / 213AH
NAME: *VMDATAL / *VMDATAH
CONTENTS: READ DATA FROM VRAM





- This is a register, which can read the data at any address of the VRAM.
- The initial address should be set by registers <2116> and <2117H>. The data can be read by the address which has been set initially.
- When reading the data continuously, the first data for the address increment should be read as "dummy" data after the address has been set.
- Quantity to be increased will be determined by "SC INCREMENT" of register <2115H> and the setting value of the "FULL GRAPHIC."

NOTE: The data can be read only during H/V BLANK or FORCED BLANK period.

(NCL PG 63)

ADDRESS: 213BH
NAME: *CGDATA
CONTENTS: READ DATA FROM CG-RAM

D7	D6	D5	D4	D3	D2	D1	D0	
CG DATA (LOW, HIGH)								
D7	(D14) D6	(D13) D5	(D12) D4	(D11) D3	(D10) D2	(D9) D1	(D8) D0	213BH

- This is a register, which can read the data at any address of the CG-RAM.
- The initial address can be set by register <2121H>. The lower 8-Bit is read first, and then the upper 7-Bit will be read by accessing the register. The current address will be increased to the next address at the same time the upper 7-Bit is read.

  Note: The data can be read only during H/V blank or forced blank period.

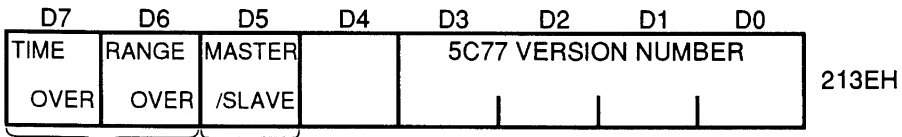
ADDRESS: 213CH / 213DH
NAME: *OPHCT / *OPVCT
CONTENTS: H/V COUNTER DATA BY EXTERNAL OR SOFTWARE LATCH

D7	D6	D5	D4	D3	D2	D1	D0	
OUTPUT DATA OF H-COUNTER								
H7	H6	H5	H4	H3	H2	H1	(H8) H0	213CH

OUTPUT DATA OF V-COUNTER								
V7	V6	V5	V4	V3	V2	V1	(V8) V0	213DH

- The H/V counter is latched by reading register <2137H>, and its H/V counter value can be read by this register.
- The H/V counter is also latched by the external latch, and its value can be read by this register.
- If register <213CH> or <213DH> is read after register <213FH> has been read, the lower 8-Bit data will be read first, and then the upper 1-Bit will be read by reading the register.

ADDRESS: 213EH
NAME: *STAT77
CONTENTS: PPU STATUS FLAG & VERSION NUMBER



MASTER / SLAVE MODE SELECT: LSI MODE (Normally "0" is set)

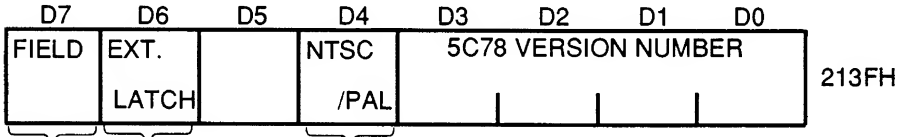
OBJ DISPLAY STATUS (ON A HORIZONTAL LINE)

RANGE: When Quantity of the OBJ (regardless of the size) becomes 33 pcs or more, "1" will be set.

TIME: When quantity of the OBJ which is converted to "8 x 8-SIZE" is 35 pcs or more, "1" will be set

NOTE: The flag will be reset at the end of the V-BLANK period.

ADDRESS: 213FH
NAME: *STAT78
CONTENTS: PPU STATUS FLAG & VERSION NUMBER



DISPLAY METHOD 0 : NTSC, 1 : PAL

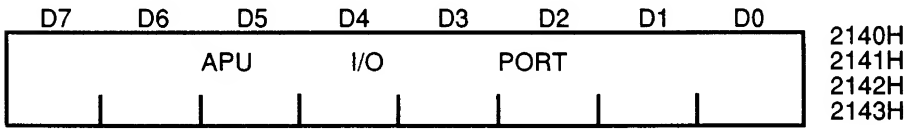
EXTERNAL LATCH FLAG: When the external signal (Light Pen, etc.) is applied, it enables to latch the H/V counter value. It is connected to I/O port d7 in SNES. (Refer to page 1-28-1.)

This is a status flag, which indicates whether the 1st or 2nd field is scanned at the interlace mode. (The definition is different from the field of NTSC.)

0 : 1ST FIELD
1 : 2ND FIELD

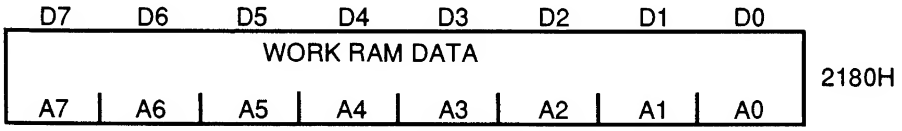
NOTE: When this register is read, registers <213CH> <213DH> will be initialized individually in the order of Low and High.

ADDRESS: 2140H / 2141H / 2142H / 2143H
NAME: APUIO0 / APUIO1 / APUIO2 / APUIO3
CONTENTS: COMMUNICATION PORT WITH APU



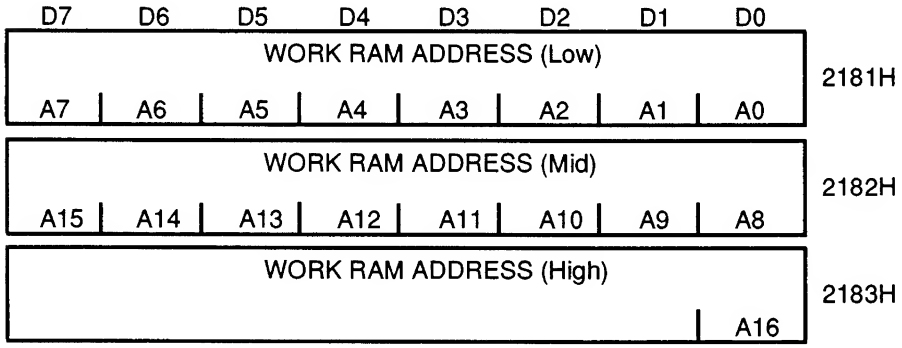
- The port provides more registers for the purpose of IN/OUT, which are 8 registers in total in the APU. Therefore, the different register will be accessed, whether reading or writing for the same address.
- Refer to Part 2 of this manual for the details of the communication method.

ADDRESS: 2180H
NAME: WMDATA
CONTENTS: DATA to consecutively read from and write to WRAM



- Data to consecutively read and write at any address of WRAM
- Data is read and written at address set by register <2181H> ~ <2183H>, and address automatically increases each time data is read or written.

ADDRESS: 2181H/2182H/2183H
NAME: WMADDL/WMADDM/WMADDH
CONTENTS: Address to consecutively read and write WRAM

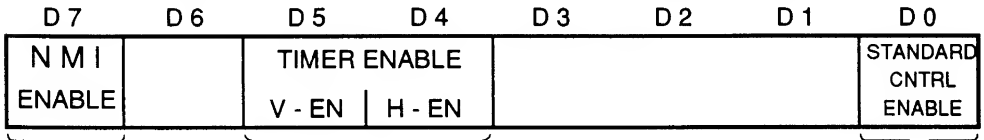


- Address to be set before WRAM is consecutively read or written.
- A0 through A16 at register <2181H> ~ <2183H> is lower 17 bit address to show address 7E0000 ~ 7FFFFF Memory.

Chapter 28. CPU Registers

ADDRESS :4200H
NAME :NMITIMEN
CONTENTS: ENABLE FLAG FOR V - BLANK, TIMER INTERRUPT & STANDARD CONTROLLER

READ



STANDARD CONTROLLER ENABLE
- 0 : Disable Automatic reading of Standard Controller
- 1 : Enable Automatic reading of Standard Controller

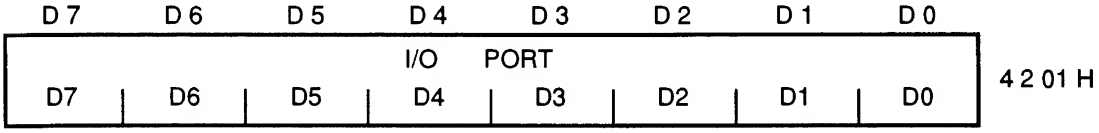
※ Reading the data can be started at the beginning of V - Blank period, but it takes about 3 or 4 scanning lines to complete the read.

TIMER ENABLE
- V - EN : V - COUNT TIMER ENABLE
- H - EN : H- COUNT TIMER ENABLE

V EN	H EN	FUNCTION
0	0	Disable BOTH H & V.
0	1	Enable H only. IRQ is applied by H - count timer value designated.
1	0	EnableV only. IRQ is applied by V - count timer value designated.
1	1	Enable both H & V. IRQ is applied by both H and V count timer value designated.

NMI ENABLE : Enable NMI at the point when V - Blank begins.
(When power is turned on or the reset signal is applied, it will be "0".)
- 0 : NMI DISABLE
- 1 : NMI ENABLE

ADDRESS :4201H
NAME :WRIO
CONTENTS: PROGRAMMABLE I/O PORT (OUT - PORT)



- This is a Programmable I/O port (OUT - PORT). The written data will be output directly from the OUT - PORT.
- When this is used as a INPORT, "1" should be written to the particular bit which will be used as a IN - PORT. The input data can be read by register <4213H>.
- Only D6 and D7 can be used by the Super NES. Standard Controller I and III (connector 1) has signal at D6 and Standard Controller II and IV (connector 2) has signal at D7. Signal at D7 is also an external latch input signal (Refer to page 1-27-23).

(NCL PG 88)

ADDRESS : 4202H / 4203H
NAME : WRMPYA / WRMPYB
CONTENTS : MULTIPLIER & MULTIPLICAND BY MULTIPLICATION

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
MULTIPLICAND - A								4 2 02 H
A7	A6	A5	A4	A3	A2	A1	A0	
MULTIPLIER - B								4 2 03 H
B7	B6	B5	B4	B3	B2	B1	B0	

- This is a register, which can set as multiplicand (A) and a multiplier (B) for Absolute Multiplication of "A (8 - bit) X B (8 - bit) = C (16 - bit)".
- A PRODUCT (C) can be read by registers <4216H><4217H>.
- Set in the order of (A) and (B). The operation will start as soon as (B) has been set, and it will be completed right after an 8-machine cycle period.
- Once the data of the A-REGISTER is set, it will not be destroyed until new data is set.

ADDRESS : 4204H / 4205H / 4206H
NAME : WRDIVL / WRDIVH / WRDIVB
CONTENTS : DIVISOR & DIVIDEND BY DIVIDE

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
DIVIDEND- C (LOW)								4 2 04 H
C7	C6	C5	C4	C3	C2	C1	C0	
DIVIDEND- C (HIGH)								4 2 05 H
C15	C14	C13	C12	C11	C10	C9	C8	
DIVISOR - B								4 2 06 H
B7	B6	B5	B4	B3	B2	B1	B0	

- This is a register, which can be set as Dividend (C) and a Divisor (B) for Absolute Divide of "C (16-bit) ÷ B (8-bit) = A (16-bit)".
- The Quotient (A) can be read by registers <4214H><4215H>. And the remainder can be read by registers <4216H><4217H>.
- Set in the order of (C) and (B). The operation will start as soon as (B) has been set, and it will be completed right after a 16-machine cycle period.
- Once the data of the C-REGISTER is set, it will not be destroyed until new data is set.

ADDRESS :4207H / 4208H
NAME :HTIMEL / HTIMEH
CONTENTS:H-COUNT TIMER SETTINGS

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
H - COUNT TIMER								
H7	H6	H5	H4	H3	H2	H1	H0	4 2 07 H
							H MSB H8	4 2 08 H

- This is a register, which can set the H-COUNT TIMER value.
- The stored value should be from 0 through 339, which is counted from the far left on the screen.
- When the coordinate counter becomes the count value set, the IRQ will be applied. And at the same time, "1" will be written to "timer IRQ" of register <4211H> (READ RESET). Enable/Disable of the interrupt will be determined by setting register <4200H>.

※ This continuous counter is reset every scanning line, therefore once the count value is stored, it is possible to apply the IRQ every time the scanning line comes to the same horizontal position on the screen.

ADDRESS :4209H / 420AH
NAME :VTIMEL / VTIMEH
CONTENTS:V-COUNT TIMER SETTINGS

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
V - COUNT TIMER								
V7	V6	V5	V4	V3	V2	V1	V0	4 2 09 H
							V MSB V8	4 2 0A H

- This is a register, which can set the V-COUNT TIMER value.
- The stored value should be from 0 through 261 (262), which is counted from top of the screen. (This line number described is different from the actual line number on the screen.)
- When the coordinate counter becomes the count value set, the IRQ will be applied. At the same time, "1" will be written to "timer IRQ" of register <4211H> (READ RESET). Enable/Disable of the interrupt will be determined by setting register <4200H>.

※ This is a continuous counter like the H-counter and will reset every time 262 lines are scanned. Once the count value is stored, it is possible to apply the IRQ every time the scanning line comes to the same vertical position on the screen.

(NCL PG 90)

ADDRESS :420BH
NAME :MDMAEN
CONTENTS:CHANNEL DESIGNATION FOR GENERAL PURPOSE DMA & TRIGGER (START)

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
GENERAL PURPOSE DMA ENABLE FLAG								4 2 0B H
CH7 EN	CH6 EN	CH5 EN	CH4 EN	CH3 EN	CH2 EN	CH1 EN	CH0 EN	

- General purpose DMA consists of 8 channels total (CH0 ~ CH7).
- This is used to designate 1 of the 8 channels (8 channels maximum).
- The channel to be used can be designated by writing "1" to the bit of this channel. As soon as "1" is written to the bit (after a few cycles have passed), the general purpose DMA transfer will begin.
- When general purpose DMA of the designated channel is completed, the flag will be cleared.

NOTE: Because the data area (register<4300H> ~) of each channel is held in common with the data of each H-DMA channel, the channel designated by the H-DMA channel designation register <420CH> can not be used. (It is prohibited to write "1" to the bit of the channel). Therefore 8 channels (CH0 ~ CH7) should be assigned by the H-DMA and the general purpose DMA.

NOTE: If the H-Blank comes during the operation of the general purpose DMA and the H-DMA is started, the general purpose DMA will be discontinued in the middle and resumed right after the H-DMA is complete.

NOTE: If 2 or more channels are designated, the DMA transfer will be performed continuously according to the priority order described on page B-1. The CPU will also stop operation until all the general purpose DMAs are completed.

ADDRESS :420CH
NAME :HDMAEN
CONTENTS:CHANNEL DESIGNATION FOR H-DMA

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
H - DMA ENABLE FLAG								4 2 0C H
CH7 EN	CH6 EN	CH5 EN	CH4 EN	CH3 EN	CH2 EN	CH1 EN	CH0 EN	

- The H-DMA consists of 8 channels total (CH0 ~ CH7).
- The register is used to designate the channel out of 8 channels (8 channels maximum).
- The channel which should be used can be designated by writing "1" to the bit of this register. As soon as H-Blank begins (after a few cycles have passed), the H-DMA transfer will begin.

NOTE: Once this flag is set, it will not be cleared until new data is set. The initial settings are done automatically for every field and the same transfer pattern will be repeated. The flag is also set out of V-Blank period, so the DMA transfer will be performed properly for the next screen frame.

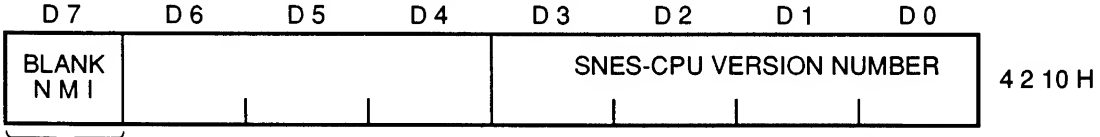
ADDRESS :420DH
NAME :MEMSEL
CONTENTS:ACCESS CYCLE DESIGNATION IN MEMORY ② AREA (Refer to "Memory Map")

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
							2.68 / 3.58	4 2 0D H

ACCESS CYCLE DESIGNATION IN MEMORY ② AREA
- 0 : 2.68MHz access cycle
- 1 : 3.58MHz access cycle (Only when the high speed memory is used)

- MEMORY ② shows the address (8000H ~ FFFFH) of the bank (80H ~ BFH) and all the addresses of the bank (C0H ~ FFH).
- When power is turned on or the reset signal is applied, it becomes "0".

ADDRESS :4210H
NAME :✶ RDNMI
CONTENTS:NMI FLAG BY V - BLANK & VERSION NUMBER



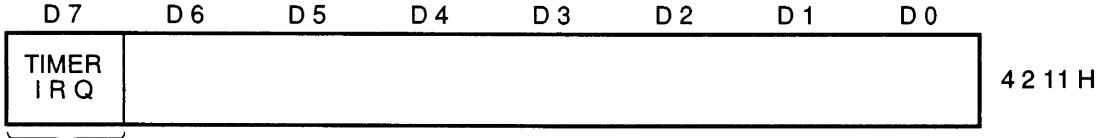
NMI FLAG BY V-BLANK : When "1" is written to "NMI ENABLE" of register <4200H>, this flag will show NMI status.

- 0 : NMI has not occurred
- 1 : NMI has occurred

✻ A "1" is written to this flag at the beginning of V-Blank and a "0" is written at the end of V-Blank. It can also be reset by reading this register.

NOTE : It is necessary to reset by reading this flag during NMI processing. (Refer to page B-3.)

ADDRESS :4211H
NAME :✶ TIMEUP
CONTENTS:IRQ FLAG BY H/V COUNT TIMER



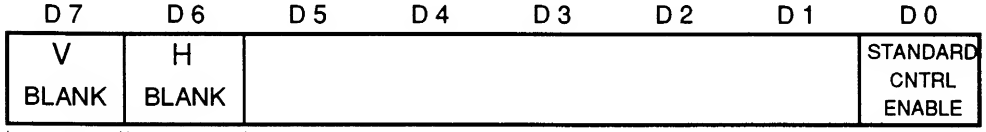
IRQ FLAG BY H/V COUNT TIMER :

This flag is "READ RESET". (If Timer Enable is set by "Timer Enable" of register <4200H>, IRQ will be applied and the flag will be set as soon as H/V count timer reaches the value stored.

✻ Even if V-EN ="0" and H-EN ="0" are set by "Timer Enable" of register <4200H>, this flag will be reset.

- 0 : Either H/V count timer is active or disabled
- 1 : Status of H/V count timer is Time-Up

ADDRESS :4212H
NAME :* HVBJOY
CONTENTS: H/V BLANK FLAG & STANDARD CONTROLLER ENABLE FLAG



STANDARD CONTROLLER ENABLE:

This flag shows the timing to read the data of the Standard Controller. (However, it is limited to the case where the "STANDARD CNTRL ENABLE" of register <4200H> is set to "1")

- 0 : This is a period when the Standard Controller is not reading the data or is disabled. (In case "0" is set to "STANDARD CNTRL ENABLE" of register <4200H>)
- 1 : This is a period when the Standard Controller is reading data.

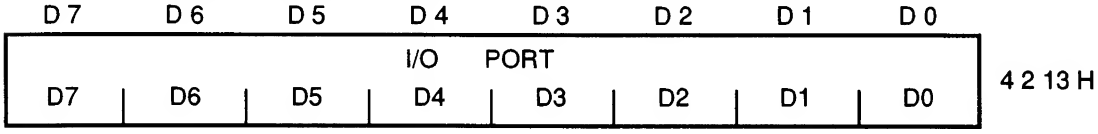
H-BLANK PERIOD FLAG : It shows whether the scan is in the H-Blank period or not.

- 0 : Out of H-Blank period
- 1 : In H-Blank period

V-BLANK PERIOD FLAG : It shows whether the scan is in the V-Blank period or not.

- 0 : Out of V-Blank period
- 1 : In V-Blank period

ADDRESS :4213H
NAME :* RDIO
CONTENTS: PROGRAMMABLE I/O PORT (IN - PORT)



- This is a Programmable I/O port (IN - PORT). The data which is set to the IN-PORT should be read directly.
- The bit in which "1" is written by register <4201H> is used as the IN-PORT.
- Only D6 and D7 can be used by the Super NES. Standard Controller I and III (connector 1) has signal at D6 and Standard Controller II and IV (connector 2) has signal at D7. Signal at D7 is also an external latch input signal (Refer to "PPU Status Flag and Version Number" in "PPU Registers").

(NCL PG 93)

ADDRESS :4214H / 4215H
NAME :✶ RDDIVL / ✶ RDDIVH
CONTENTS:QUOTIENT OF DIVIDE RESULT

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
QUOTIENT - A (LOW)								4 214 H
A7	A6	A5	A4	A3	A2	A1	A0	
QUOTIENT - A (HIGH)								4 215 H
A15	A14	A13	A12	A11	A10	A9	A8	

- This is Quotient (A), which is a result of absolute division of " C (16 BIT) ÷ B (8 BIT) = A (16 BIT)".
- Dividend (C) and divisor (B) are set by registers <4204H>, <4205H>, and <4206H>.

ADDRESS :4216H / 4217H
NAME :✶ RDMPYL / ✶ RDMPYH
CONTENTS:PRODUCT OF MULTIPLICATION RESULT OR REMAINDER OF DIVIDE RESULT

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
PRODUCT-C [MULTIPLICATION] / REMAINDER [DIVIDE] (LOW)								4 216 H
C7	C6	C5	C4	C3	C2	C1	C0	
PRODUCT-C [MULTIPLICATION] / REMAINDER [DIVIDE] (HIGH)								4 217H
C15	C14	C13	C12	C11	C10	C9	C8	

① WHEN USED FOR MULTIPLICATION

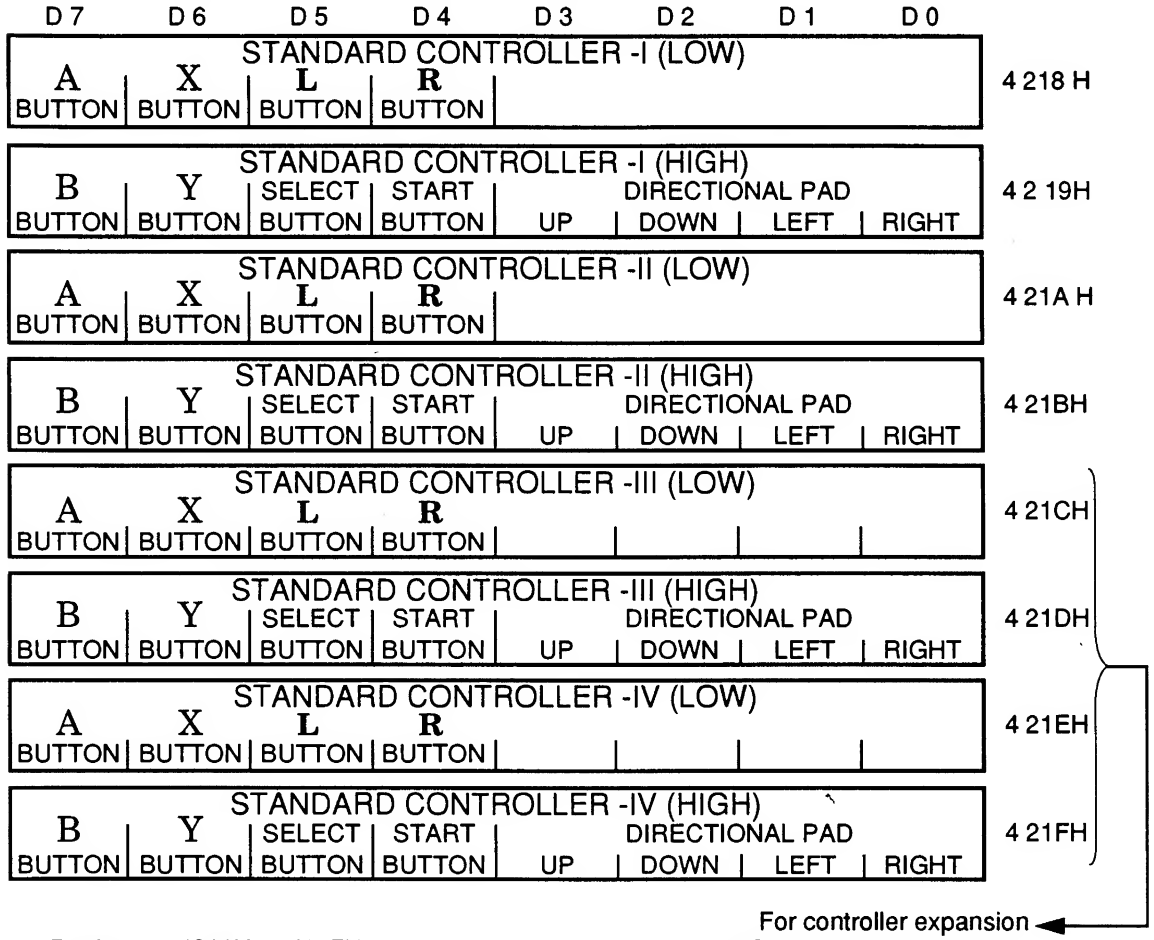
- This is a Product (C), which is a result of Absolute Multiplication of " A (8 BIT) X B (8 BIT) =C (16 BIT)".
- Multiplicand (A) and Multiplier (B) are set by registers <4202H> and <4203H>.

② WHEN USED FOR DIVISION

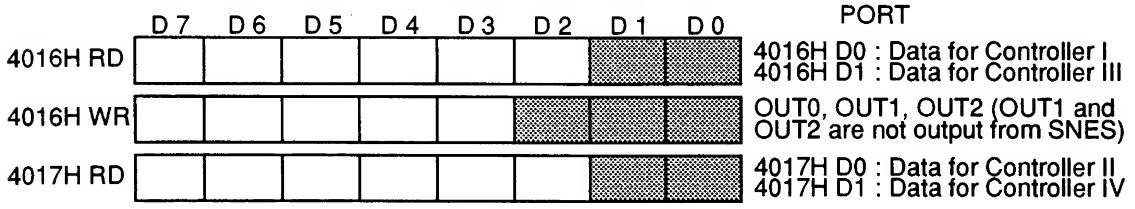
- This is a Remainder, which is a result of Absolute Division of " C (16 BIT) ÷ B (8 BIT) = A (16 BIT) • • • REMAINDER (8 or 16 bit)".
- Dividend (C) and divisor (B) are set by registers <4204H>, <4205H>, and <4206H>.

(NCL PG 94)

ADDRESS :4218H/4219H/421AH/421BH/421CH/421DH/421EH/421FH
NAME :STD CNTRL1L / 1H / 2L/2H/3L/3H/4L/4H
CONTENTS:DATA FOR STANDARD CONTROLLER I, II, III, & IV



- Registers <4016H> <4017H> can be used the same as the NES.

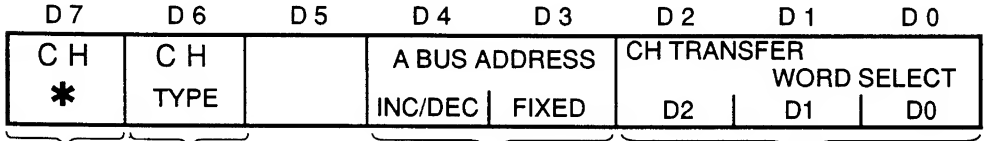


NOTE : Whether the standard controllers are connected to Super NES unit or not can be determined by reading the 17th bit of 4016H and 4017H. (Refer to "Standard Controller".)

1 : connected
0 : not connected

(NCL PG 95)

ADDRESS : 43X0H (X : CHANNEL NUMBER <0 ~ 7>)
NAME :
CONTENTS : PARAMETER FOR DMA TRANSFER



* Transfer Origination

DMA TRANSFER WORD SELECT

GENERAL PURPOSE DMA : B-ADDRESS CHANGE METHOD DESIGNATION PER CHANNEL

D2	D1	D0	ADDRESS TO BE WRITTEN
0	0	0	1-ADDRESS
0	0	1	2-ADDRESS (VRAM etc.) L,H
0	1	0	1-ADDRESS (WRITE TWICE)
0	1	1	2-ADDRESS (WRITE TWICE) L,L,H,H
1	0	0	4-ADDRESS L,H,L,H

H-DMA : The number of bytes to be transferred per line and write method designation.

D2	D1	D0	# OF BYTE TO BE TRANSFERRED	ADDRESS TO BE WRITTEN
0	0	0	1 BYTE	1-ADDRESS
0	0	1	2 BYTE	2-ADDRESS (VRAM etc.) L,H
0	1	0	2 BYTE	1-ADDRESS (WRITE TWICE) L,L
0	1	1	4 BYTE	2-ADDRESS (WRITE TWICE) L,L,H,H
1	0	0	4 BYTE	4-ADDRESS L,H,L,H

FIXED ADDRESS FOR A-BUS & AUTOMATIC INCREMENT /DECREMENT SELECT [IN CASE OF GENERAL PURPOSE DMA]

D3
- 0 : Automatic address increment/decrement
- 1 : Fixed address (To be used when clearing VRAM etc.)

D4
- 0 : Automatic increment
- 1 : Automatic decrement

(In case "0" is written to D3)

TYPE DESIGNATION [H-DMA ONLY] : Addressing mode designation when accessing the data (Refer to page B-2).
- 0 : ABSOLUTE ADDRESSING
- 1 : INDIRECT ADDRESSING

TRANSFER ORIGINATION DESIGNATION : Transfer direction A Bus→B Bus B Bus→ A Bus Designation (Refer to page B-1)
- 0 : A-BUS →B-BUS (CPU MEMORY→PPU)
- 1 : B-BUS →A-BUS (PPU→ CPU MEMORY)

* For example, in case the DMA transfer is performed from CPU memory to PPU, "0" should be written.

(NCL PG 96)

ADDRESS :43X1H (X : CHANNEL NUMBER <0 ~ 7>)
NAME :
CONTENTS:B-BUS ADDRESS FOR DMA

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
B - ADDRESS								43X1H
BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0	

- This is a register which can set the address of B-bus.
- Whether this is the address of the "Transfer Destination" or the address of the "Transfer Origination" can be determined by D7 (Transfer Origination) of register <43X0H>.

A - BUS ← Direction can be designated by "Transfer Origination" → B - BUS
Actual address is 0021XXH.
(XX : value by this register)

※ When H-DMA is performed, it will be the address of the "Transfer Destination".

ADDRESS :43X2H / 43X3H / 43X4H (X : CHANNEL NUMBER <0 ~ 7>)
NAME :
CONTENTS:TABLE ADDRESS OF A-BUS FOR DMA <A1 TABLE ADDRESS>

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
A1 TABLE ADDRESS (LOW)								43X2H
A7	A6	A5	A4	A3	A2	A1	A0	
A1 TABLE ADDRESS (HIGH)								43X3H
A15	A14	A13	A12	A11	A10	A9	A8	
A-TABLE BANK								43X4H
A23	A22	A21	A20	A19	A18	A17	A16	

- This is a register, which can set the address of A-bus.
- Whether this is the address of the "Transfer Destination" or the address of the "Transfer Origination" can be determined by D7 (Transfer Origination) of register <43X0H>.
  A "0" should be written to D7 except in special cases.
- In the H-DMA mode, the address of the transfer origination is designated except it is a special case. Therefore, for the CPU area designated by this address, the data (page B-2) must be set by the absolute addressing mode or the indirect addressing mode.
- This address becomes the basic address on the A-Bus during DMA transfer period and the address will be increased or decreased based on this address. (When the general purpose DMA is performed, it will be decreased.)

ADDRESS : 43X5H / 43X6H / 43X7H (X : CHANNEL NUMBER <0 ~ 7>)
NAME :
CONTENTS : DATA ADDRESS STORE BY H-DMA
& NUMBER OF BYTE TO BE TRANSFERRED SETTINGS BY GENERAL PURPOSE DMA

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
DATA ADDRESS (LOW)								FOR H-DMA
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	43X5H
NUMBER OF BYTES TO BE TRANSFERRED (LOW)								GENERAL PURPOSE DMA
B7	B6	B5	B4	B3	B2	B1	B0	
DATA ADDRESS (HIGH)								FOR H-DMA
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	43X6H
NUMBER OF BYTES TO BE TRANSFERRED (HIGH)								GENERAL PURPOSE DMA
B15	B14	B13	B12	B11	B10	B9	B8	
DATA BANK								FOR H-DMA
DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	43X7H

- IN CASE OF H-DMA
  This is a register in which the indirect address will be stored automatically in the Indirect addressing mode. The indirect address means the data described on page B-2. It is not necessary to read or write directly by the CPU except in special cases.

- IN CASE OF GENERAL PURPOSE DMA
  This is the register which can set the number of bytes to be transferred. However, the number of Byte (0000H) means 10000H.

ADDRESS :43X8H / 43X9H (X : CHANNEL NUMBER <0 ~ 7>)
NAME :
CONTENTS:TABLE ADDRESS OF A-BUS BY DMA <A2 TABLE ADDRESS>

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
A2 TABLE ADDRESS (LOW)								43X8H
A7	A6	A5	A4	A3	A2	A1	A0	
A2 TABLE ADDRESS (HIGH)								43X9H
A15	A14	A13	A12	A11	A10	A9	A8	

- This is the address which is used to access the CPU and RAM. It will be increased automatically. (See page B-2.)
- The data of this register is used as the basic address which is the address set by the "A1 Table Address". Afterwards, because it will be increased (or decreased) automatically, it is not necessary to set the address into this register by the CPU directly.

(However, if the data which is transferred needs to be changed by force, it can be done by setting the CPU memory address to this register. In such case, the address of the CPU which is accessed currently will be changed by reading this register.) } H-DMA ONLY

ADDRESS :43XAH (X : CHANNEL NUMBER <0 ~ 7>)
NAME :
CONTENTS:THE NUMBER OF LINES TO BE TRANSFERRED BY H-DMA

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
NUMBER OF LINES								43XAH
Continue	L6	L5	L4	L3	L2	L1	L0	

- This is the register which shows the number of lines for H-DMA transfer. (Refer to page B-2.)
- The number of lines written to the CPU memory will be the basic number of lines. It is not necessary to write the data into this register by the CPU directly.

Chapter 1. SNES Sound Source Outline

1.1 OUTLINE

The SNES sound source is composed of a Sound-CPU-IC, a single chip in which are integrated an 8-bit CPU, IPL ROM, I/O ports, a DSP-IC, and peripheral apparatus.

CHARACTERISTICS

•CPU core	: Sony SPC700 series CMOS 8-bit CPU
•Minimum Command Execution Time	: 1.953μs/2.48MHz when active
•Internal ROM	: 64 byte (IPL ROM)
•Memory Space	: 64K byte
•Peripheral Functions	
•I/O Ports	: SNES CPU Interface I/O Ports 8 bit x 4
Universal I/O Ports 8 bit x 2	
•Timers	: (8 bit timer + 4 bit counter) x 3 sets
•Output Sound Production	: 4-bit ADPCM sampling sound x 8 tones (simultaneous production)

(NCL PG 2)

1.2 SYSTEM OUTLINE

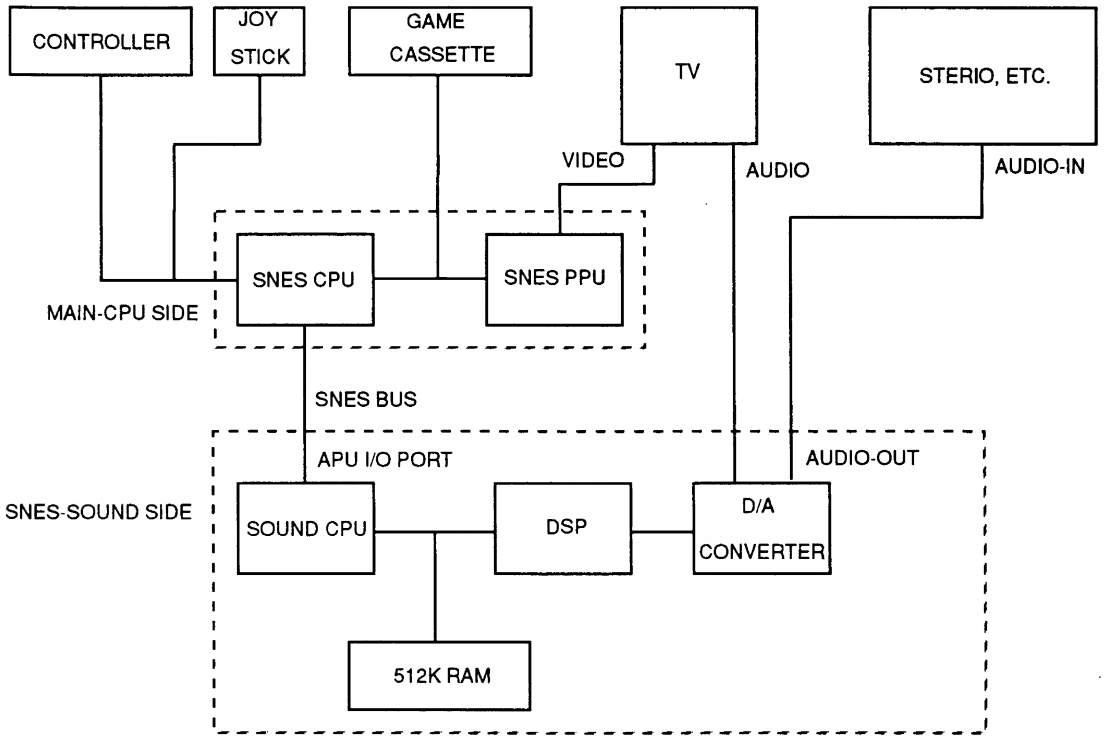


Figure 3-1-1 System Block Diagram

1.3 Designation and Role of Each Section:

1.3.1 Sound-CPU:

SNES sound source CPU. Program and tone color data are read into RAM from the game cassette through the SNES CPU, Consequently controlling the game music.

In addition, the Sound- CPU is provided with an internal IPL-ROM which is activated upon reset. The IPL-ROM provides for transmission of data through the SNES CPU, initial settings of the SNES sound source, etc.

1.3.2 DSP:

Digital Signal Processor. Reproduces tone quality data in RAM. Carries out various functions for the purpose of musical expression.

1.3.3 512K RAM:

Shared on a time basis by the Sound-CPU and the DSP.

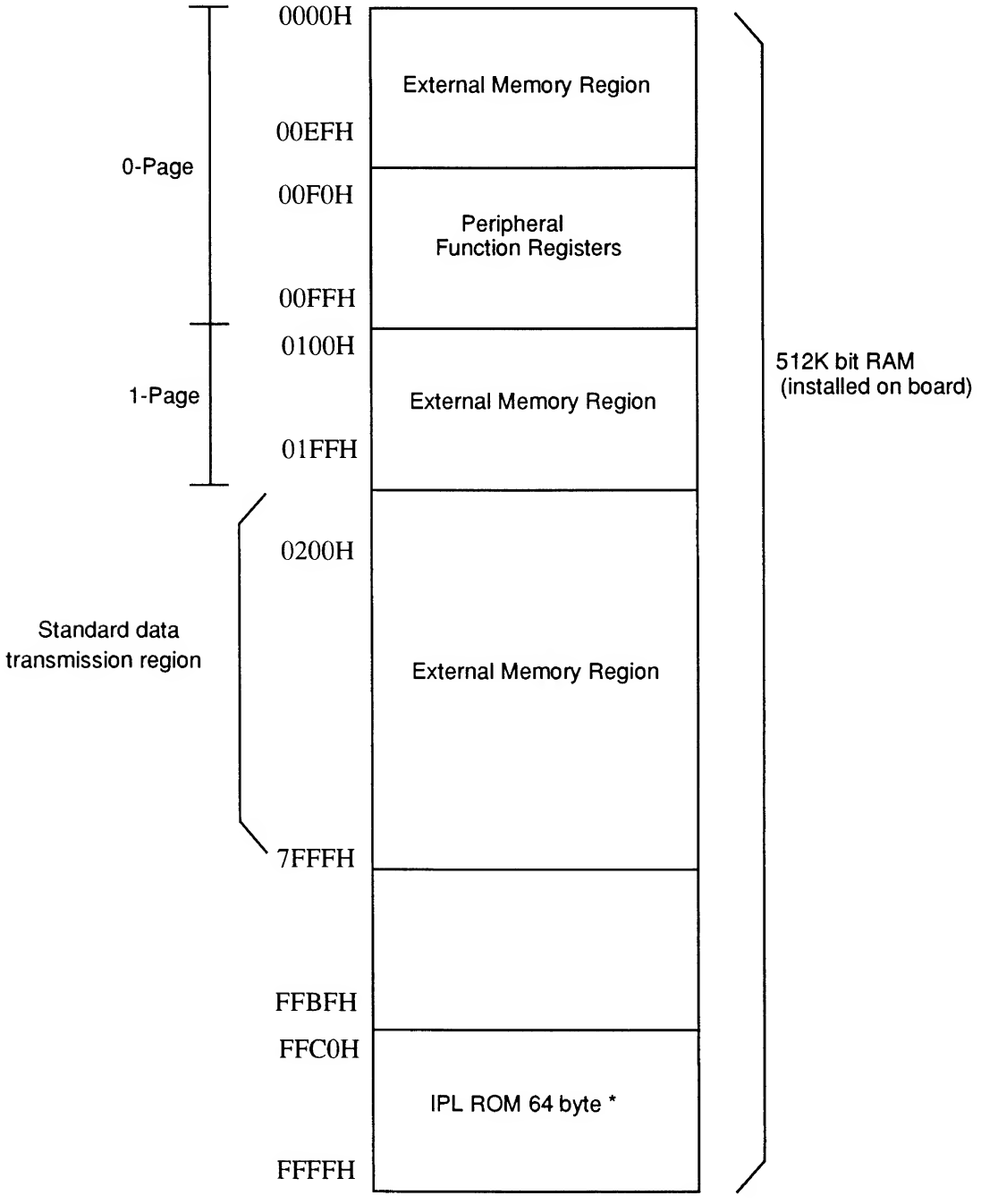
1.3.4 SNES CPU:

CPU for SNES use. Carries out progression of the game in conformity with the game cassette format.

1.3.5 SPPU:

PPU for SNES use. Creates imaging through CPU control.

1.4 MEMORY MAPPING



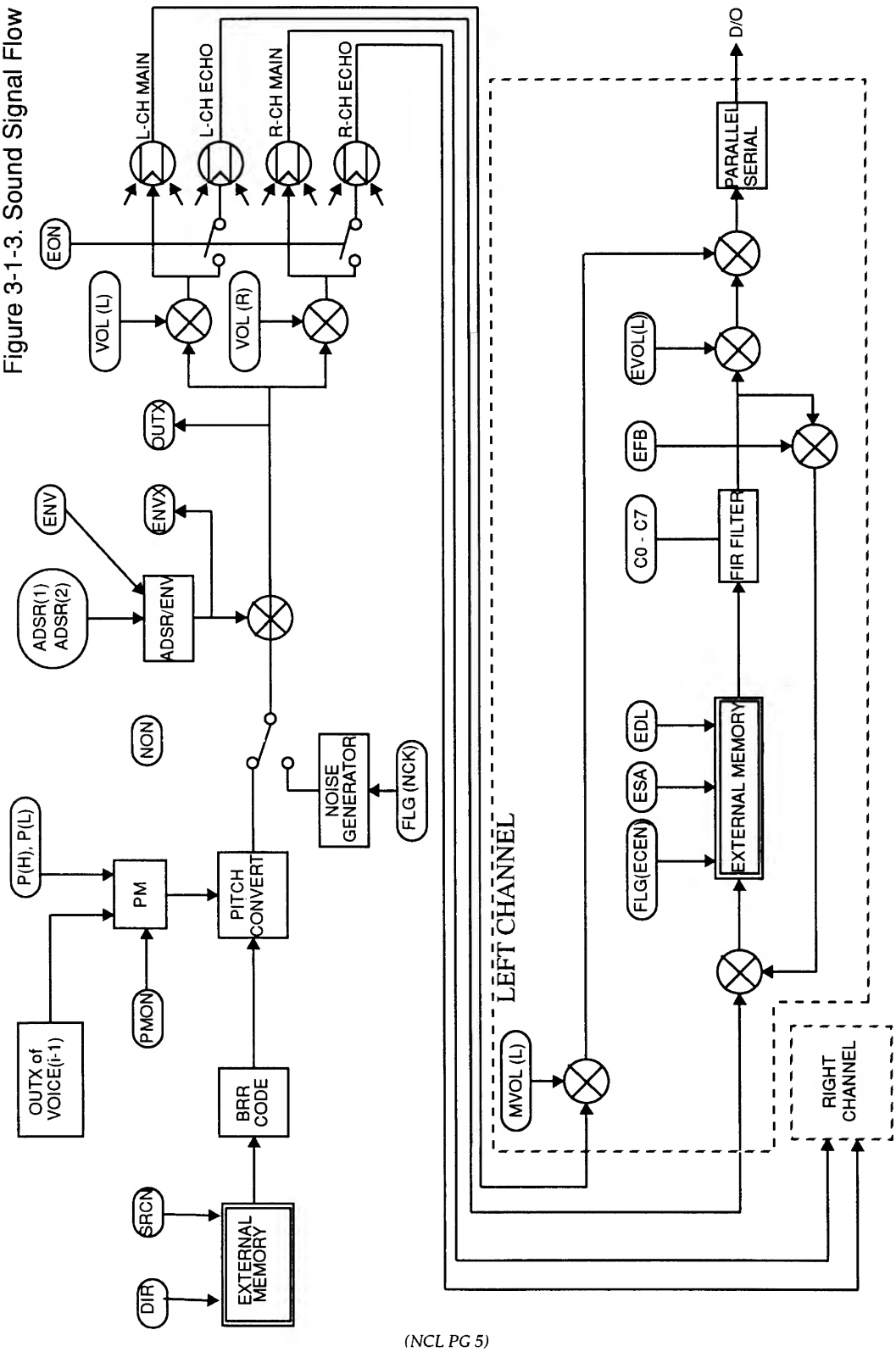
* The initial hardware setting program is installed in the IPL ROM

Figure 3-1-2 Memory Map

(NCL PG 4)

1.5 SIGNAL FLOW

Figure 3-1-3. Sound Signal Flow



(NCL PG 5)

Chapter 2. BRR (Bit Rate Reduction)

Sound data for the Super NES is recorded on a game data cassette in 4-bit ADPCM format. Creating data in this format requires the use of a technique called "BRR" for some sounds.
Knowledge of this operation is not necessary to correctly create data with the Sony NEWS system, which has been the standard tool used by Nintendo to date. When the same data is created using a different tool, however, one must understand BRR.
Complications arise during the creation of sound data from data in BRR format when creating the position of the program which selects the filter number described below. Also, Nintendo cannot currently support this programming effort.
One block of wave form data is comprised of a one-byte header and eight-byte wave form (4 bit x 16 samples). This is the minimum unit the sound IC can handle.

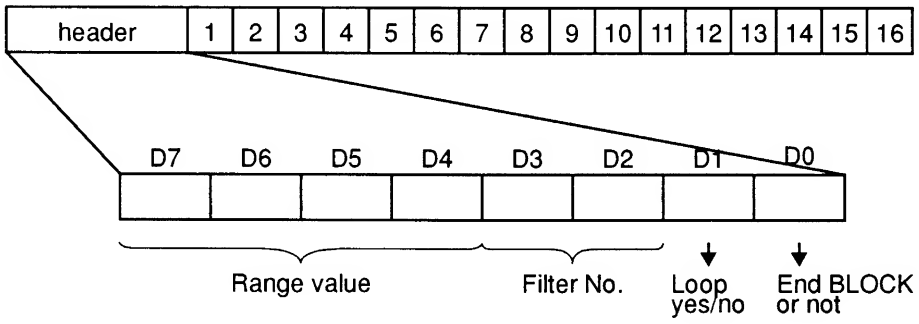


Figure 3-2-1 BRR Data String

From this, the range and filter contain the BRR information (how the data string in 1 - 16 will be regenerated). Refer to ¶ 7.2.2.5 for information on d1 and d0 bits.
The "filter number" can have a value from 0 ~ 3. These indicate, respectively, 0: constant, 1: first-order filter, 2: second-order filter, 3: third-order filter. Of these values, 1 ~ 3 require previous data values for data calculation. (Filter 1 requires one previous sample, and filters 2 and 3 require two previous samples.)
The "range" indicates the number of bits shifted. One sample data (4-bit) is shifted to the left for the number of bits and re-created as 16-bit data. The maximum value for a range is 12(1100). (See the next page.)

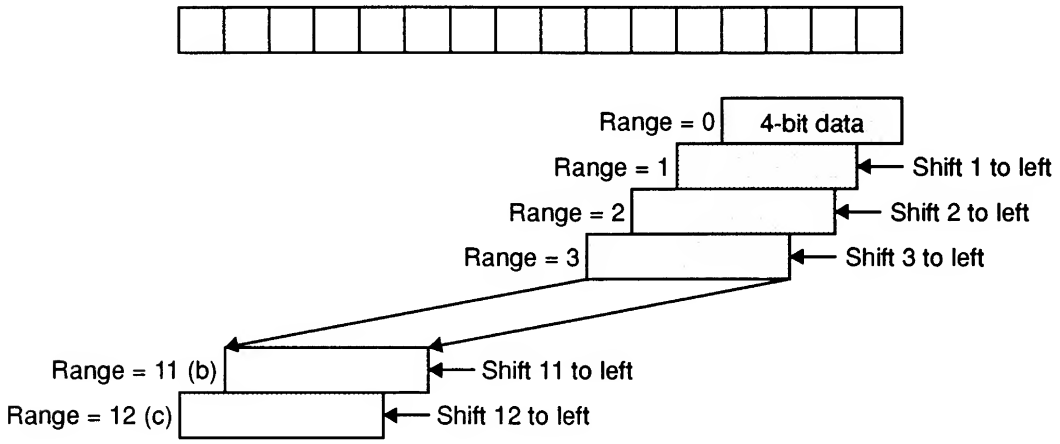


Figure 3-2-2 BRR Range Data

An equation for decoding a sample value, *x* from the previous sample x_{-1} and second previous sample x_{-2}, is given below.

$$x = R + ax_{-1} + bx_{-2}$$

R is the value obtained by shifting the 4-bit data, *d*, by the range value, *r*.

$$R = [d]\ 2^{r-15}$$

([d] is a decimal presentation of *d*, which is in two's compliment form, -7 ~+8)

The values of *a* and *b* for each filter are as follows:

Filter No.	a	b
0	0	0
1	0.9375	0
2	1.90625	-0.9375
3	1.796875	-0.8125

Table 3-2-1 BRR Filter Values

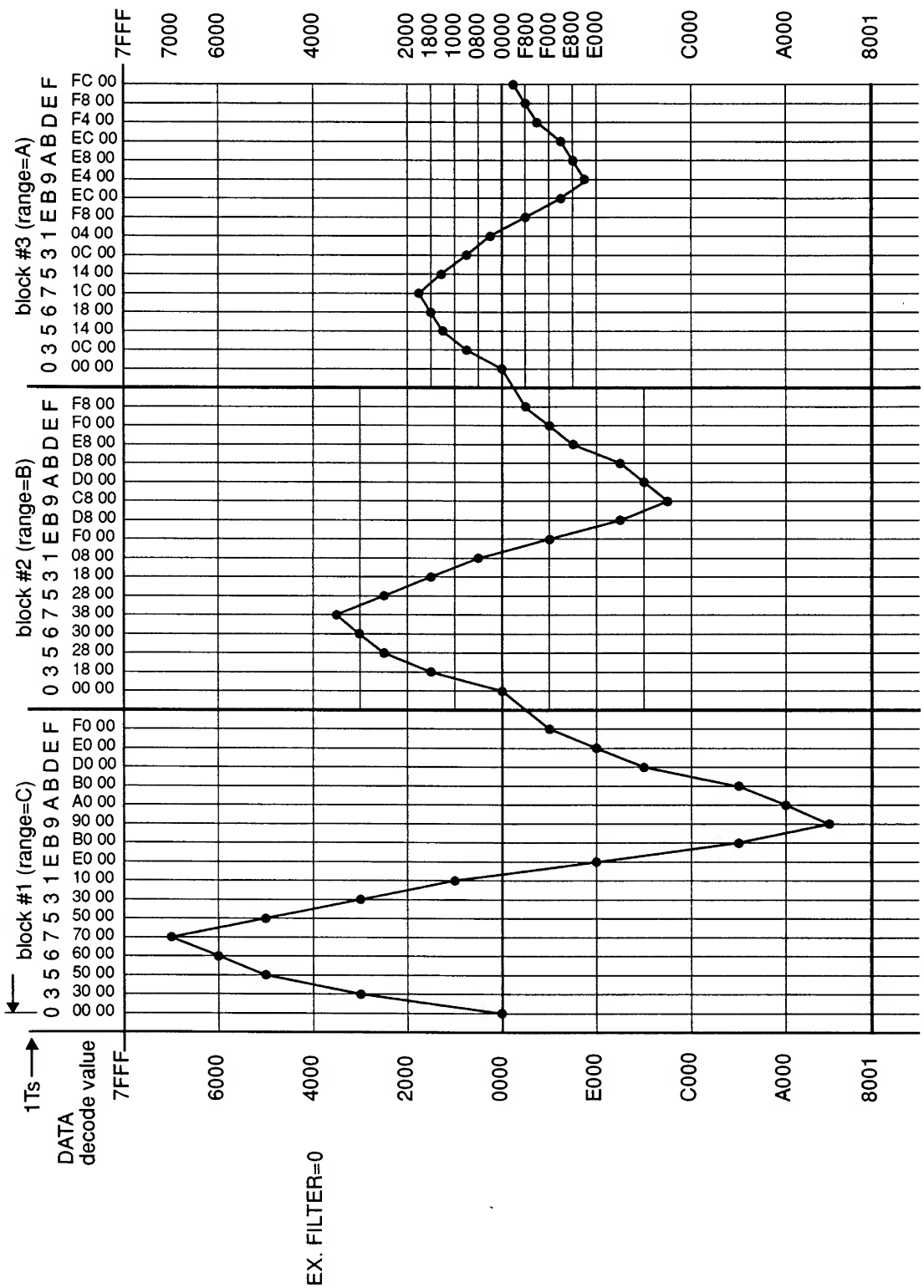


Figure 3-2-3 Example Data when Filter = 0.

Chapter 3. I/O Ports

3.1 PERPHERAL FUNCTIONS REGISTERS

Peripheral Function Registers Table 3-3-1. Peripherals

Address	Function Register	R/W	When Reset	Remarks
00FOH	(test)	---	----------	Installed in Sound-CPU
00F1H	Control	W	Control = "--00-000"	
00F2H	Register Add.	R/W	Indeterminate	Installed in DSP
00F3H	Register Data	R/W	Indeterminate	Installed in DSP
00F4H	Port-0	R/W	Port 0r = "00" Port 0w = "00"	Installed in Sound-CPU
00F5H	Port-1	R/W	Port 1r = "00" Port 1w = "00"	Installed in Sound-CPU
00F6H	Port-2	R/W	Port 2r = "00" Port 2w = "00"	Installed in Sound-CPU
00F7H	Port-3	R/W	Port 3r = "00" Port 3w = "00"	Installed in Sound-CPU
00F8H	-----	---	----------	----------
00F9H	-----	---	----------	----------
00FAH	Timer-0	W	Indeterminate	Installed in Sound-CPU
00FBH	Timer-1	W	Indeterminate	Installed in Sound-CPU
00FCH	Timer-3	W	Indeterminate	Installed in Sound-CPU
00FDH	Counter-0	R	Indeterminate	Installed in Sound-CPU
00FEH	Counter-1	R	Indeterminate	Installed in Sound-CPU
00FFH	Counter-3	R	Indeterminate	Installed in Sound-CPU

(NCL PG 6)

3.2 APU I/O PORTS

Ports 0-3 are ports which carry out data transmission to the SCPU through the SNES bus and are composed of four 8-bit input registers and four 8-bit output registers. Port *n* r registers can only write from the SCPU section and can only read from the Sound-CPU section. The opposite is true of the port *n* w registers. Since the composition of each of these ports is identical, only an explanation of Port 0r and Port 0w is provided.

1. Data is input into Port 0r when the SCPU writes data into 2140H. Then, the contents of Ports 0r are read when the Sound-CPU reads the data in 00F4H (this is also true of Ports 1r - 3r).
2. Data is written into Port 0w when the Sound-CPU writes data into the APU I/O port (00F4H). Then the contents of Port 0w are read when the SCPU reads 2140H (this is also true of Ports 1w - 3w).
3. When reset is applied, the contents of Port *n* r registers and Port *n* w registers become "00" (*n* =0-3).

Table 3-3-2 Port0 - Port3 Registers

Address Seen From Sound-CPU	Address Seen From SCPU	Register Name	W/R	Function Seen From Sound-CPU Section
00F4H	2140H	Port0r Port0w	R W	Read content of Port0r register. Write to Port0w register.
00F5H	2141H	Port1r Port1w	R W	Read content of Port1r register. Write to Port1w register.
00F6H	2142H	Port2r Port2w	R W	Read content of Port2r register. Write to Port2w register.
00F7H	2143H	Port3r Port3w	R W	Read content of Port3r register. Write to Port3w register.

Figure 3-3-1 I/O Diagram

(NCL PG 7)

Chapter 4. Control Register

4.1 THE PORT CLEAR FUNCTION BY MEANS OF THE CONTROL REGISTER.

The ports are cleared to "00" when "1" is written into the control register port clear control bits PC32 and PC10. When "0" is written in, they are not cleared.
When "1" is written into the port clear control bit PC10, both the port 0r register and the port 1r register are cleared to "00". In the same manner, when "1" is written into PC32, both the port 2r register and the port 3r register are cleared to "00".

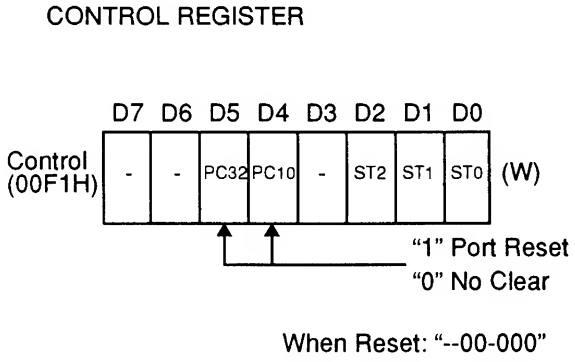
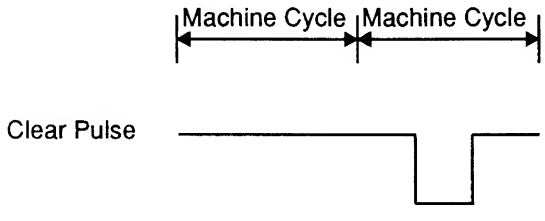


Figure 3-4-1 Port Clear

Note: Clear Timing
Port clear is executed during the machine cycle following that in which "1" is written into the port clear control bit.
When port clear timing conflicts with write timing to the port in question from the SNES bus, there are cases in which the contents of the register in question become indeterminate.



Timing when "1" is written into the port clear control bit.

Figure 3-4-2 Clear Timing

(NCL PG 8)

4.2 TIMER CONTROL BY MEANS OF THE CONTROL REGISTER

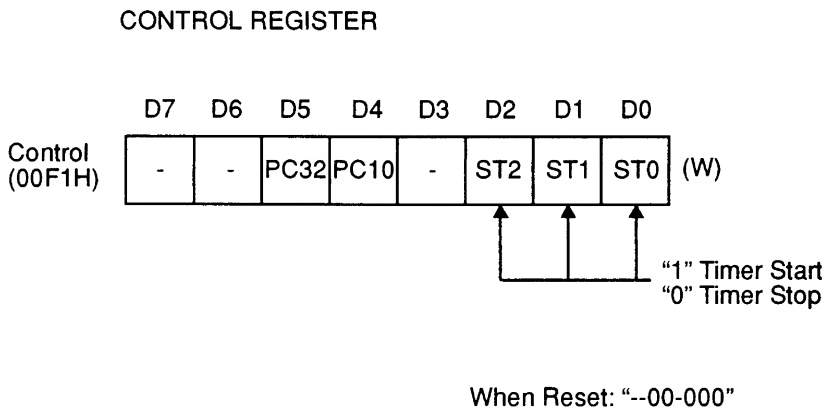


Figure 3-4-3 Timer Control

ST0 is the Timer T0 start/stop control bit; the timer stops with "0" and starts with "1". At this timer, it is necessary to input "1" into ST0 once it has been changed to "0".
ST1 and ST2 are respectively the start/stop control bits of timers T1 and T2. Their function is identical to that of ST0.
NOTE: In regard to the functional operation of timers, please refer to the next page.

(NCL PG 9)

Chapter 5. Timers

5.1 FUNCTION OF TIMERS T0, T1, AND T2

The SNES sound source is provided with three timers; T0, T1, and T2.

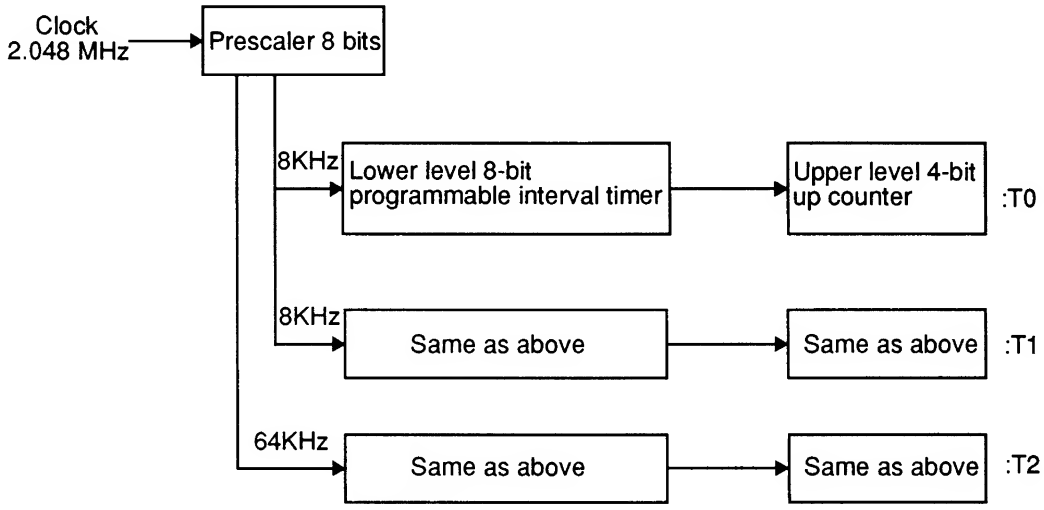


Figure 3-5-1 Timer Section

The timers T0, T1, and T2 are each composed of a lower level 8-bit programmable interval timer connected to an upper level 4-bit up counter.

The 8-bit timer is made up of an 8-bit binary up counter, comparator, timer register, and control circuit. Each of the timers; T0, T1, and T2; is independently programmable.

The clock input to timers T0 and T1 from the prescaler is 8KHz (125 µs) and the clock input to timer T2 from the prescaler is 64KHz (15.6 µs).

	8-bit Timer		4-bit Up Counter
	Resolution	Max. Count Value	Max. Count Value
Timer T0, T1	125 µsec.	32 msec.	512 msec.
Timer T2	15.6 µsec.	4 msec.	64 msec.

Table 3-5-1 Timer Function

(NCL PG 10)

5.2 TIMER ACTION

Since timers T0, T1, and T2 are alike in structure, an explanation of only timer T0 is provided.

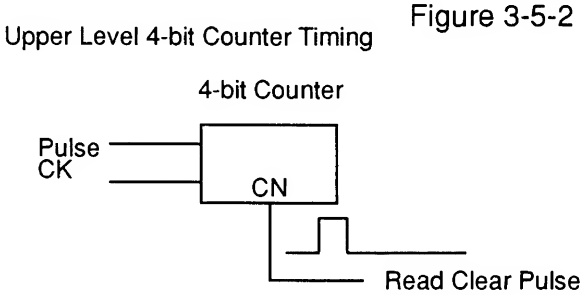
The lower level 8-bit timer of timer T0 is composed principally of a binary up counter, which is incremented at each count of the clock input. When its value corresponds to the contents of the timer register, it is cleared to 00H. Simultaneously a pulse is generated to the 4-bit up counter.

The 4-bit up counter is composed principally of a binary up counter, which increments at each input of a lower level pulse.

The action of the counter of timer T0 is controlled by the 0 bit of the control register. When bit ST0 is "0" count up is suspended. Count up commences when both upper and lower level counters are cleared by "1". Consequently, in order to clear the counters, it is necessary to set bit ST0 to "1" after having set it to "0".

Writing to the timer register is carried out while the counter is stopped. At this time the minimum write value is 00H and the maximum value is 01H. Though it is not possible to read the value of the timer register, it is possible to read the 4-bit value CN0 at any time. When the value of CN0 is read, only the 4-bit up counter section is cleared to "00".

Figure 3-5-2 4-bit Counter



Action of timer T0 is stopped by means of the reset input (POR="L"). At the time of reset, ST0 of the control register is "0" and; CN0 and TM0 of the timer register are indeterminate.

When CN is read, the 4-bit up counter alone is cleared through IC internal timing, But the read clear pulse and the pulse to the 4-bit up counter do not conflict with each other.

Consequently, when the pulse is input to the 4-bit up counter, the value of CN will necessarily be incremented; or when the value of CN is read, CN will be cleared and become "0".

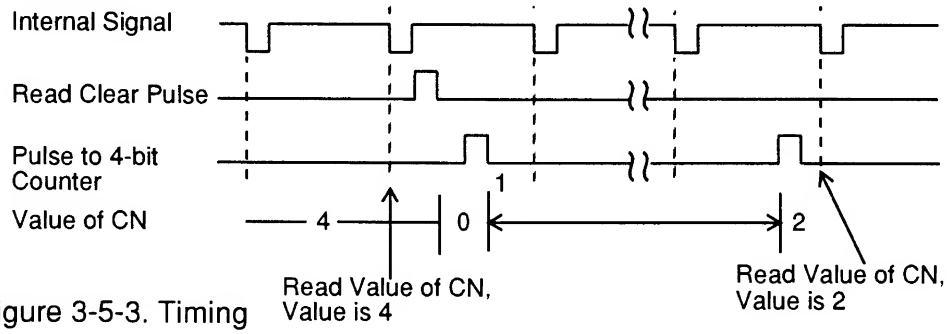
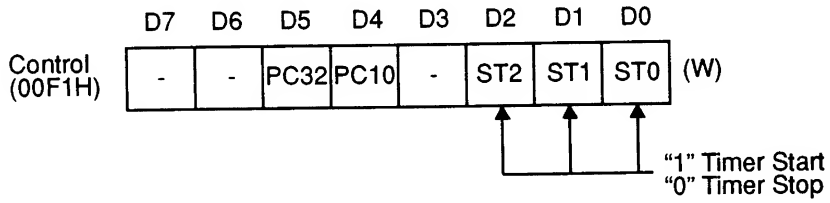


Figure 3-5-3. Timing

(NCL PG 11)

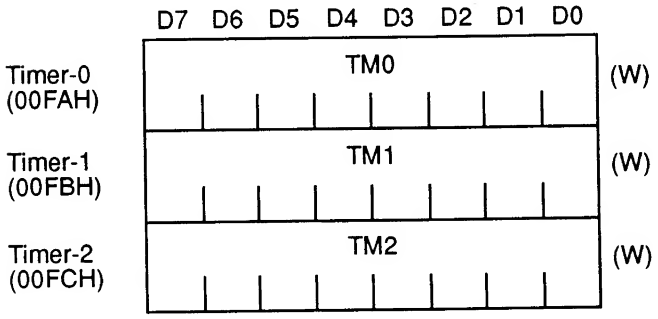
5.3 TIMER RELATED REGISTERS

Control Register

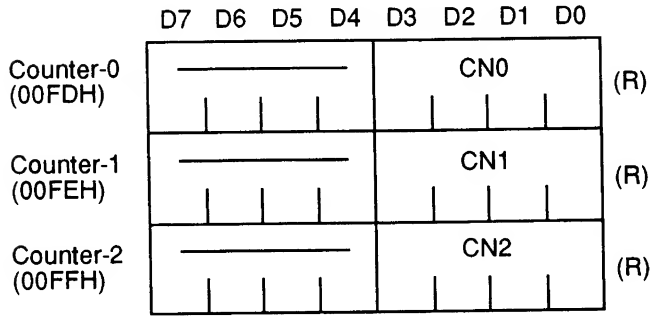


When Reset: "--00-000"

Timer Register



Counter Register



Indeterminate
When Reset

Figure 3-5-4 Timer Related Registers

(NCL PG 12)

Chapter 6. DSP Interface Register

6.1 Interface Register

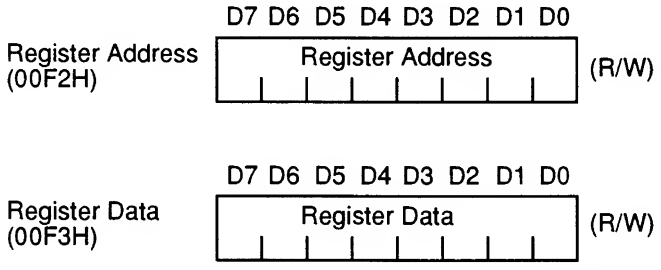
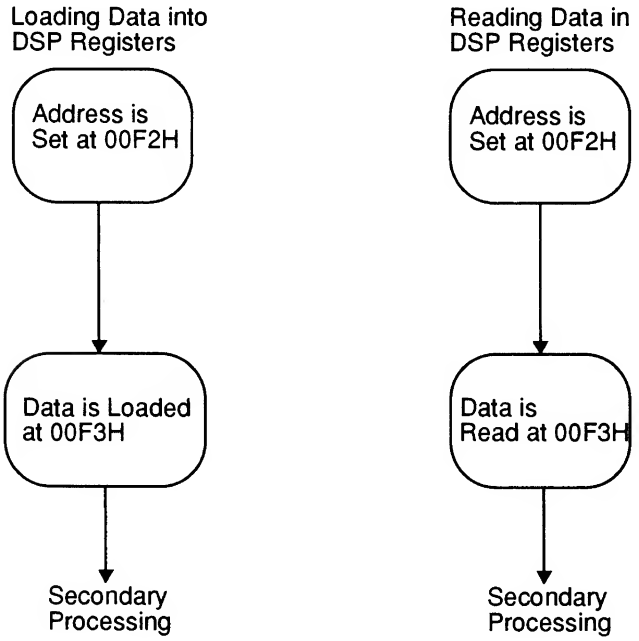


Figure 3-6-1. Interface Register

This is the register which loads data into the registers within DSP. Values are loaded into the designated register in accordance with the path of the flow-chart below. The DSP address is written to 00F2H and data is written to 00F3H (refer to Flow A). When the contents of the register data is read, it conforms to Flow B. The address to be read is loaded into 00F2H and the contents of 00F3H are read.

Figure 3-6-2. Interface Register Flow



(NCL PG 13)

Chapter 7. Register Used

7.1 DSP REGISTER MAP

Address	Register	Explanation of Function
00	VOL (L)	Left Channel Volume **VOICE 0**
01	VOL (R)	Right Channel Volume
02	P (L)	The total 14 bits of P(H) and P(L) express
03	P (H)	Pitch Height
04	SRCN	Designates source number from 0-255
05	ADSR (1)	Address is designated by D7=1 of ADSR(1); when
06	ADSR (2)	D7=0, Gain is operative
07	GAIN	Envelope can be freely designated by the program
08	∴ENVX	Present value of envelope which DSP rewrites at each Ts
09	∴OUTX	Value after envelope multiplication & before VOL multiplication (present wave height value)
10 ~ 19	Voice 1	Same as Voice 0
20 ~ 29	Voice 2	
30 ~ 39	Voice 3	
40 ~ 49	Voice 4	
50 ~ 59	Voice 5	
60 ~ 69	Voice 6	
70 ~ 79	Voice 7	
0C	MVOL (L)	Main Volume (L)
1C	MVOL (R)	Main Volume (R)
2C	EVOL (L)	Echo Volume (L)
3C	EVOL (R)	Echo Volume (R)
4C	KON	Key On, D0-D7 correspond to Voice0-Voice7
5C	KOF	Key Off
6C	FLG	Designated on/off of reset, mute, echo, and noise clock
7C	∴ENDX	Indicates source end block
0D	EFB	Echo Feedback
1D	---	Not Used
2D	PMON	Pitch modulation of Voice i with OUTX of Voice (i-1) as modulated wave
3D	NON	Noise on/off, D0-D7 correspond to Voice0-Voice7
4D	EON	Echo On/Off
5D	DIR	Off-set address of source directory
6D	ESA	Off-set address of echo region, Echo Start Address
7D	EDL	Echo Delay. Only lower level 4 bits active.
0F	Filter Coefficients C0	Echo Filter coefficients Makes up an 8 tap FIR Filter (Both Lch & Rch have the same filter)
1F	C1	
2F	C2	
3F	C3	
4F	C4	
5F	C5	
6F	C6	
7F	C7	

∴ Register written to by DSP during conditions of activity.

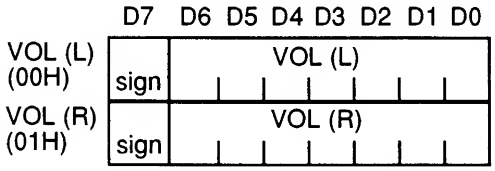
Table 3-7-1 DSP Register Map

(NCL PG 14)

7.2 REGISTER FUNCTION

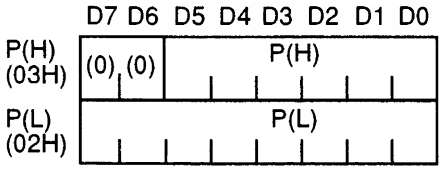
7.2.1 Register of each voice (Addresses indicated are those of Voice 0).

7.2.1.1 VOL (L), VOL (R)



Each is a volume level multiplied by Lch and Rch, which is in a 2's complement form making D7 the sign bit. When a negative value is entered, phases reverse.

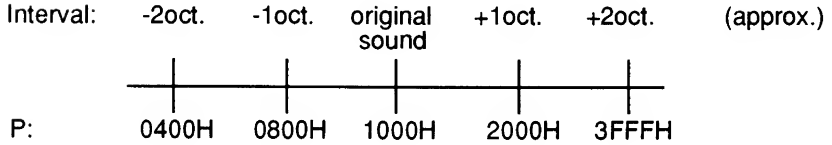
7.2.1.2 P(L), P(H)



Pitch is expressed by the total 14 bits combining six lower level bits of P(H) and eight bits of P(L). At the current time, two upper level bits of P(H) are not used. (Considered to be "0" at all times.) With f **as the frequency of the reproduced sound, f_0 as the frequency of the original sound (sound at the time of recording), and P as the value expressed by the lower level fourteen bits of P(H) and P(L), the following formula is performed:**

$$f = f_0 \bullet \frac{P}{2^{12}}$$

The diagram below illustrates the relationship between P and the octaval ratio of the reproduced sound and the original sound. There are theoretically no limitations in the practical range so long as the original sound is converted **lower**. The upper range is limited to approximately four times the frequency of the original sound.



In terms of tone quality, the lower level 4 bits of P(L) should be set at "0" when possible in cases where pitch aberrations are not of concern.

7.2.1.3 ADSR(1), ADSR(2)

	D7	D6	D5	D4	D3	D2	D1	D0
ADSR(1) (05H)	ADSR /GAIN	DR			AR			
ADSR(2) (06H)	SL				SR			

When D7 of ADSR(1) = "1", these two bytes become operable. (ADSR mode)
AR is added to the fixed value "1/64" and DR, SR by the fixed value "1-1/256". When in the state of "Key Off", the "click" sound is prevented by the addition of the fixed value "1/256". (GAIN mode is identical.)

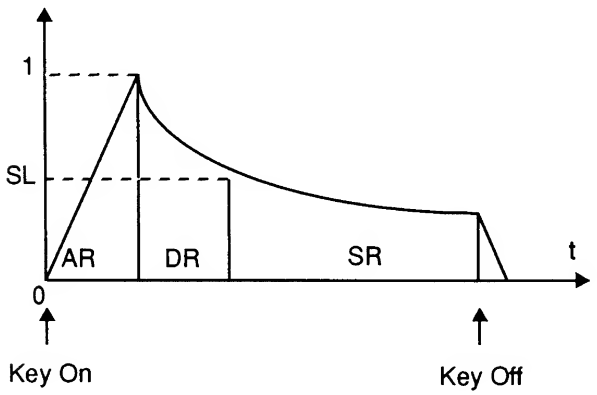
Table 3-7-2 Adsr Parameters

AR	Time from 0 to 1
0	4.1 sec
1	2.6 sec
2	1.5 sec
3	1.0 sec
4	640 msec
5	380 msec
6	260 msec
7	160 msec
8	96 msec
9	64 msec
A	40 msec
B	24 msec
C	16 msec
D	10 msec
E	6 msec
F	0 msec

DR	Time from 1 to SL
0	1.2 sec
1	740 msec
2	440 msec
3	290 msec
4	180 msec
5	110 msec
6	74 msec
7	37 msec

SL	Ratio
0	1/8
1	2/8
2	3/8
3	4/8
4	5/8
5	6/8
6	7/8
7	1

SR	Time from 0 to 1
0	Infinite
1	38 sec
2	28 sec
3	24 sec
4	19 sec
5	14 sec
6	12 sec
7	9.4 sec
8	7.1 sec
9	5.9 sec
A	4.7 sec
B	3.5 sec
C	2.9 sec
D	2.4 sec
E	1.8 sec
F	1.5 sec
10	1.2 sec
11	880 msec
12	740 msec
13	590 msec
14	440 msec
15	370 msec
16	290 msec
17	220 msec
18	180 msec
19	150 msec
1A	110 msec
1B	92 msec
1C	74 msec
1D	55 msec
1E	37 msec
1F	18 msec



7.2.1.4 GAIN

This becomes operable when D7 of ADSR(1) = 0. The following five modes are available.

	D7	D6	D5	D4	D3	D2	D1	D0
Direct Designation (07H)	0							
Increase Mode (Linear) (07H)	1	1	0					
Increase Mode (Bent Line) (07H)	1	1	1					
Decrease Mode (Linear) (07H)	1	0	0					
Decrease Mode (Exponential) (07H)	1	0	1					

∴ **Direct Designation: The value of GAIN is set directly by the values of D0 ~ D6.**

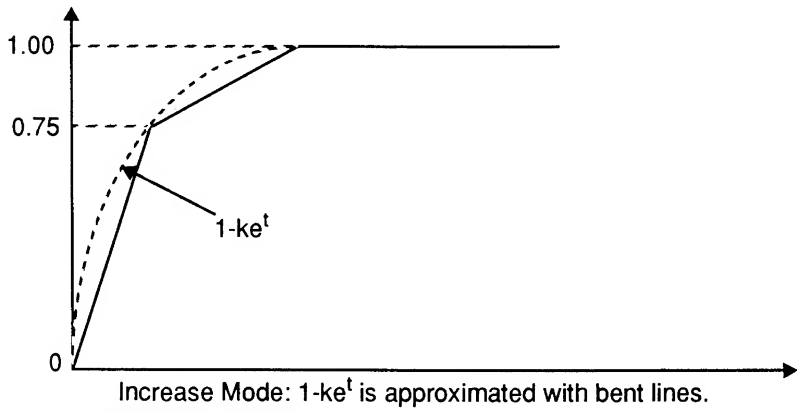
∴ **Increase (Linear): Addition of the fixed value 1/64.**

∴ **Increase (Bent Line): Addition of the constant 1/64 up to 0.75, addition of the constant 1/256 from 0.75 to 1.**

∴ **Decrease (Linear): Subtraction of the fixed value 1/64.**

∴ **Decrease (Exponential): Multiplication by the fixed value 1-1/256.**

In all cases, present envelope values (indicated by ENVX) are utilized for initial values.



Increase Mode: $1-ke^t$ is approximated with bent lines.

Figure 3-7-1 Bent Line Mode

The various parameter values are indicated on the next page.

GAIN PARAMETERS

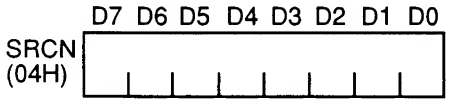
Parameter Values	Increase Mode Linear (0→1)	Increase Mode Bentline (0→1)	Decrease Mode Linear (1→0)	Decrease Mode Exponential (0→1/10)
00	Infinite	Infinite	Infinite	Infinite
01	4.1 sec	7.2 sec	4.1 sec	38 sec
02	3.1 sec	5.4 sec	3.1 sec	28 sec
03	2.6 sec	4.6 sec	2.6 sec	24 sec
04	2.0 sec	3.5 sec	2.0 sec	19 sec
05	1.5 sec	2.6 sec	1.5 sec	14 sec
06	1.3 sec	2.3 sec	1.3 sec	12 sec
07	1.0 sec	1.8 sec	1.0 sec	9.4 sec
08	770 msec	1.3 sec	770 msec	7.1 sec
09	640 msec	1.1 sec	640 msec	5.9 sec
0A	510 msec	900 msec	510 msec	4.7 sec
0B	380 msec	670 msec	380 msec	3.5 sec
0C	320 msec	560 msec	320 msec	2.9 sec
0D	260 msec	450 msec	260 msec	2.4 sec
0E	190 msec	340 msec	190 msec	1.8 sec
0F	160 msec	280 msec	160 msec	1.5 sec
10	130 msec	220 msec	130 msec	1.2 sec
11	96 msec	170 msec	96 msec	880 msec
12	80 msec	140 msec	80 msec	740 msec
13	64 msec	110 msec	64 msec	590 msec
14	48 msec	84 msec	48 msec	440 msec
15	40 msec	70 msec	40 msec	370 msec
16	32 msec	56 msec	32 msec	290 msec
17	24 msec	42 msec	24 msec	220 msec
18	20 msec	35 msec	20 msec	180 msec
19	16 msec	28 msec	16 msec	150 msec
1A	12 msec	21 msec	12 msec	110 msec
1B	10 msec	18 msec	10 msec	92 msec
1C	8 msec	14 msec	8 msec	74 msec
1D	6 msec	11 msec	6 msec	55 msec
1E	4 msec	7 msec	4 msec	37 msec
1F	2 msec	3.5 msec	2 msec	18 msec

Table 3-7-3Gain Parameters

(NCL PG 18)

7.2.1.5 SRCN

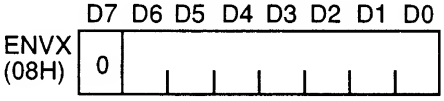
Refers to source number. It is the sequence of tone color within the hexa-file of tones produced by means of a separate tool. (0 ~ 255)



7.2.1.6 ENVX

The present value of the ADSR/GAIN envelope constant. The DSP section rewrites this at each Ts (31.25 μsec).

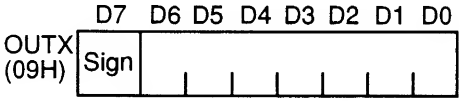
Seven bits without a sign bit. (D7 is always 0).



7.2.1.7 OUTX

The present value of the wave height after envelope multiplication and prior to VOL multiplication. The DSP section rewrites this at each Ts. (31.25 μsec). Its value is utilized as the modulated wave of pitch modulation.

Eight bits with a sign bit.



7.2.2 COMPLETE VOICE REGISTERS

7.2.2.1 KON, KOF

"Key on" and "Key off". D0 ~ D7 correspond to Voice 0 ~ 7. When a "1", key on or key off are active; when "0" neither is active. These two registers need not be reset. With KOF, in regard to any Voice in which a "1" is written and whether in the ADSR mode or GAIN mode, 1 to 0 decreases at the rate of 8 msec by means of the addition of the fixed value 1/256. In writing in a succession of KON and KOF, two Ts (62.5 μsec) or more should be released. (In writing a succession of various data in less that 2 Ts, the data written may not be usable later.)

	D7	D6	D5	D4	D3	D2	D1	D0
KON (4CH)	Voice 7	Voice 6	Voice 5	Voice 4	Voice 3	Voice 2	Voice 1	Voice 0
KOF (5CH)	Voice 7	Voice 6	Voice 5	Voice 4	Voice 3	Voice 2	Voice 1	Voice 0

7.2.2.2 PMON

Pitch modulation is imposed on Voice ***n*** with OUTX of Voice(***n***-1) (***n***=1-7) as a modulated wave. When D***n***=1, it becomes modulation ON. (For example, when D1=1, a modulated tone is generated from Voice 1.) However modulation does not affect Voice 0. Therefore, the bit D0 is not active. In regard to the method of pitch modulation, when y_0 is the wave height value of the modulated wave and P is the value of P(H) and P(L), then:

$$P' = P\ (1+y_0)$$

The value of P", as above, is substituted for P and used as the pitch at that time.

	D7	D6	D5	D4	D3	D2	D1	D0
PMON (2DH)	Voice 7	Voice 6	Voice 5	Voice 4	Voice 3	Voice 2	Voice 1	

7.2.2.3 NON

Noise on/off. D0 ~ 7 correspond to Voice 0 ~ 7. When on, noise is issued instead of sound source data. At this time, if sound source data of formants only is designated through the previous SRCN, then noise is generated only for the duration of the sound source data. When reproduction for random lengths of time is desired, sound source data incorporating a loop must be designated through the SRCN. In addition, even though two or more Voices may be on, the source of noise is the same.

Note: Modulation can not be imposed on this noise.

	D7	D6	D5	D4	D3	D2	D1	D0
NON (3DH)	Voice 7	Voice 6	Voice 5	Voice 4	Voice 3	Voice 2	Voice 1	Voice 0

(NCL PG 20)

7.2.2.4 EON

Echo on/off. Active "1". D0 ~ 7 correspond to Voice 0 ~ 7.

	D7	D6	D5	D4	D3	D2	D1	D0
EON (4DH)	Voice 7	Voice 6	Voice 5	Voice 4	Voice 3	Voice 2	Voice 1	Voice 0

(5) FLG

	D7	D6	D5	D4	D3	D2	D1	D0
FLG (6CH)	RES	MUTE	$\overline{\text{ECEN}}$	NCK				

RES: Soft reset is turned on when D7=1. At this time, all Voices are in a state of "Key On" suspension and Mute is turned on. It becomes a "1" with power on.

MUTE: Mute is turned on in all Voices when D6=1. This always occurs when power is first applied.

$\overline{\text{ECEN}}$: Allows the possibility to write into external memory through Echo, when D5=0. (Echo Enable). After power on, read out data is indeterminate until initial data is written in by the CPU.

NCK: Designates the clock of the noise generator.

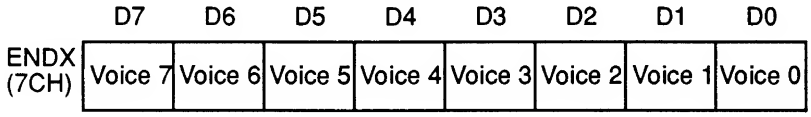
Table 3-7-4. Noise Generator Clock

NCK	Freq.	NCK	Freq.	NCK	Freq.	NCK	Freq.
00	0 Hz	08	83 Hz	10	500 Hz	18	3.2 KHz
01	16 Hz	09	100 Hz	11	667 Hz	19	4.0 KHz
02	21 Hz	0A	125 Hz	12	800 Hz	1A	5.3 KHz
03	25 Hz	0B	167 Hz	13	1.0 KHz	1B	6.4 KHz
04	31 Hz	0C	200 Hz	14	1.3 KHz	1C	8.0 KHz
05	42 Hz	0D	250 Hz	15	1.6 KHz	1D	10.7 KHz
06	50 Hz	0E	333 Hz	16	2.0 KHz	1E	16 KHz
07	63 Hz	0F	400 Hz	17	2.7 KHz	1F	32 KHz

It is only possible to write into these registers from the CPU section.

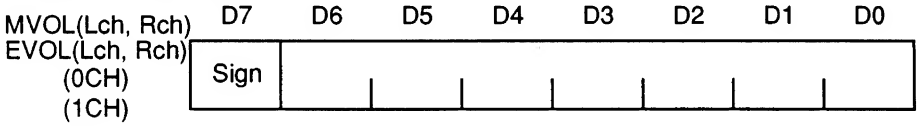
7.2.2.5 ENDX

When BRR decode of the block having the Source End flag is completed, the DSP section sets up a "1". D0 ~ 7 correspond to Voice 0 ~ 7. If there is a voice which has been keyed on, the bit corresponding to this voice is reset. In addition, when the CPU section writes into this register, all bits are reset.



7.2.2.6 MVOL(L), MVOL(R), EVOL(L), and EVOL(R)

Refer to Main Volume (Lch, Rch) and Echo Volume (Lch, Rch). The output of this register is the sum of main volume and echo volume with a sign bit.

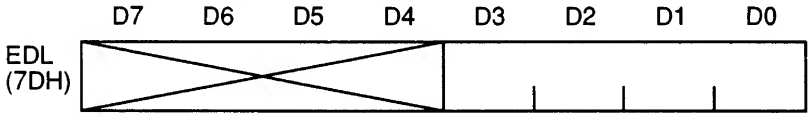


7.2.2.7 ESA

Echo Start Address. Issues the off-set address of the Echo region. (ESA) x 100H becomes the lead-off address of the Echo region.

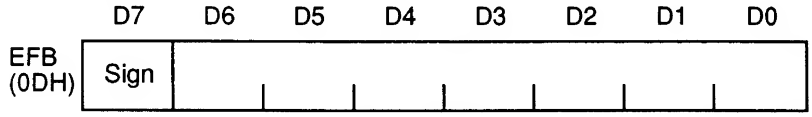
7.2.2.8 EDL

Echo Delay. Only the lower level four bits are used. Delay time o is an interval of 16 msec. and is variable within a range of 0 ~ 240 msec. If this time is considered to be ***t***, the necessary external memory region is (2***t***) Kbytes, with a maximum allowable of 30 Kbytes. However, when EDL=0, the four byte memory region of ESA - ESA+3 becomes necessary.



7.2.2.9 EFB

Refers to Echo Feed-Back. This word consists of eight bits including a sign bit.

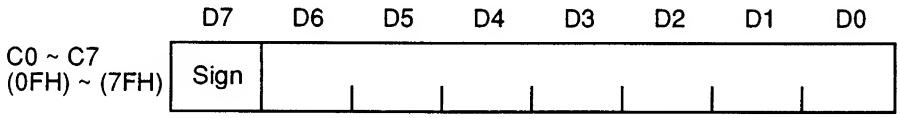


7.2.2.10 DIR

Issues the off-set address of the source directory. (DIR) x 100Hs is the beginning address of the directory.

7.2.2.11 C0 ~ C7

Issues the filter coefficient. It is composed of eight bits, including a sign bit and makes up an eight tap FIR filter (identical with that of Lch and Rch).



Filter Setting Example 1: When a low pass filter is imposed on the echo sound.

Register	Numerical Value
C0	FF
C1	08
C2	17
C3	24
C4	24
C5	17
C6	08
C7	FF

Filter Setting Example 2: When the echo sound is given the same tone color as the original sound.

Register	Numerical Value
C0	7F
C1	00
C2	00
C3	00
C4	00
C5	00
C6	00
C7	00

7.3 SOUND SOURCE DATA (SOURCE) SPECIFICATIONS *

Sound source data is produced according to the following specifications by means of specialized tools.

7.3.1 Source Directory

7.3.1.1 SA(H), SA(L)

The source start address. This 16 bit address is the lead-off address of the lead-off block.

7.3.1.2 LSA(H), LSA(L)

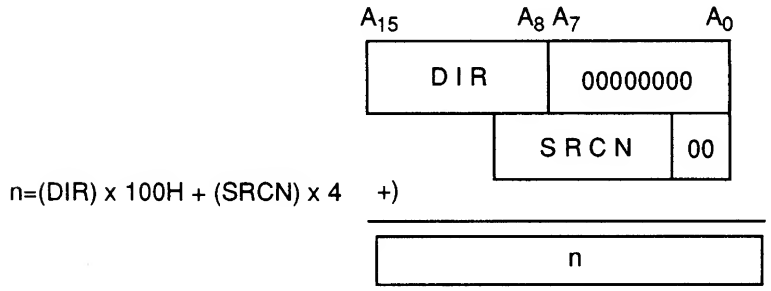
Source loop start address. This 16 bit address is the lead-off address of the loop start block.

Table 3-7-5. Source Directory

Memory Address	Directory
n+0	SA(L)
n+1	SA(H)
n+2	LSA(L)
n+3	LSA(H)

SA: Source Start Address

LSA: Source Loop Start Address



* Sound source data used in the SNES is called "Source".

7.3.2 SOURCE DATA

7.3.2.1 BLOCK FORMAT

The sound, sampled at 32KHz, undergoes BRR (bit rate reduction) processing and the data is condensed from 16 bits to 4 bits. The four-bit data is arranged into sixteen portions and, together with the RF register, is formed into one block of nine bytes.

	D7	D6	D5	D4	D3	D2	D1	D0
RF	BRR Data						Loop on/off	END
D_{A0}	$D_{A0}H$				$D_{A0}L$			
D_{B0}	$D_{B0}H$				$D_{B0}L$			
D_{A1}	$D_{A1}H$				$D_{A1}L$			
D_{B1}	$D_{B1}H$				$D_{B1}L$			
D_{A2}	$D_{A2}H$				$D_{A2}L$			
D_{B2}	$D_{B2}H$				$D_{B2}L$			
D_{A3}	$D_{A3}H$				$D_{A3}L$			
D_{B3}	$D_{B3}H$				$D_{B3}L$			

Table 3-7-6 Source Data Block Format

7.3.2.2 RF

Bits D7 ~ D2 is composed of data relating to BRR. When D1=1, it indicated that it is a source having a loop and when D1=0, it indicates that the block is the block with the final data.

Chapter 8. CPU Organization

A Sony SPC700 series is used in the CPU core of the SNES Sound Source. It is possible to access and address space of 64 Kbytes in the SPC series CPU. Address classification of the memory space is made according to purpose; addresses 0000_H ~ $00FF_H$ are called page 0 and addresses 0100_H ~ $01FF_H$ are called page 1. In regard to the data in this region; when direct page designation is carried out by the direct page flag (P) within the program status word, it is possible to carry out data processing in wide-ranging addressing modes with a small number of machine cycles.

Within the CPU there are the universal registers A, X, and Y, program status word (PSW) of the various flags, program counter (PC), and stack pointer (SP).

The A register is operable by the greatest number of commands and becomes an 8-bit operation accumulator. When 16-bit operations are carried out, it becomes paired with the Y register and becomes the lower level 8-bit register of the 16-bit accumulator. The X and Y registers, in addition to their function as universal registers, are used in various operations. These include; functions as index register of various index addressing modes, as dual address command source, destination address register, etc.
In the command set there are single address commands which carry out arithmetic and logical operations centered in the A register and dual address commands which can designate random addresses within the direct page as source addresses and destination addresses.

In regard to bit processing diversified by control purpose, Boolean bit operation commands are applicable to the 8 Kbyte wide range of data of addresses 0000_H ~ $1FFF_H$. Moreover, in regard to the bits within the direct page, set, reset and bit conditional relative jump can be utilized. In regard to the data within the total space of the 64 Kbytes; commands of multiple bit test and set, test, and reset are provided. For the purpose of data which must be systematized or in order to carry out data processing rapidly, it is possible to operate 16-bit data with a single command. Addition, subtraction, comparison, and transference are possible between two bytes of continuous 16-bit data within the direct page and the paired Y register and A register. In addition, increment and decrement of continuous 16-bit data within the direct page are possible.

There are multiplication and division commands for the purpose of rapid data processing and processing of data in a variety of forms. Multiplication is 8-bits x 8-bits with no sign and is carried out with the multiplicand stored in the Y register and the multiplier stored in the A register; the result is entered into the (Y,A) 16-bit accumulator. Division is 16 bits/8 bits with no sign and is carried out with the dividend stored in the (Y,A) 16 bit accumulator and the divisor stored in the X register. The resulting quotient is entered into the A register and the remainder into the Y register.

When processing decimal data, there are decimal addition/subtraction correcting commands in regard to the results of both addition and subtraction.

In regard to branched commands, there are relative branched commands according to the conditions of the various status flags, according to the conditions of set or reset of random bits within the direct page, etc. In addition, in regard to looped branched commands, there are comparison branched commands and subtraction branched commands. For these there are two types of addressing modes.

In regard to subroutine call commands, there are subroutine address direct designation, Three-byte call commands within the 64 Kbytes, Two-byte call commands for calling specific areas, and One-byte call commands using call tables. It is possible to improve byte efficiency through proper usage in response to the frequency of subroutine use.

8.1 CPU REGISTERS

Within the CPU are the registers necessary for the execution of various commands. These are the A register (also functions as an 8-bit accumulator), X register, Y register (8-bit universal register which can also be used as an index register), PSW (program status word), SP (stack pointer), etc. These are all 8-bit registers, but the PC (program counter) is made up of 16 bits.

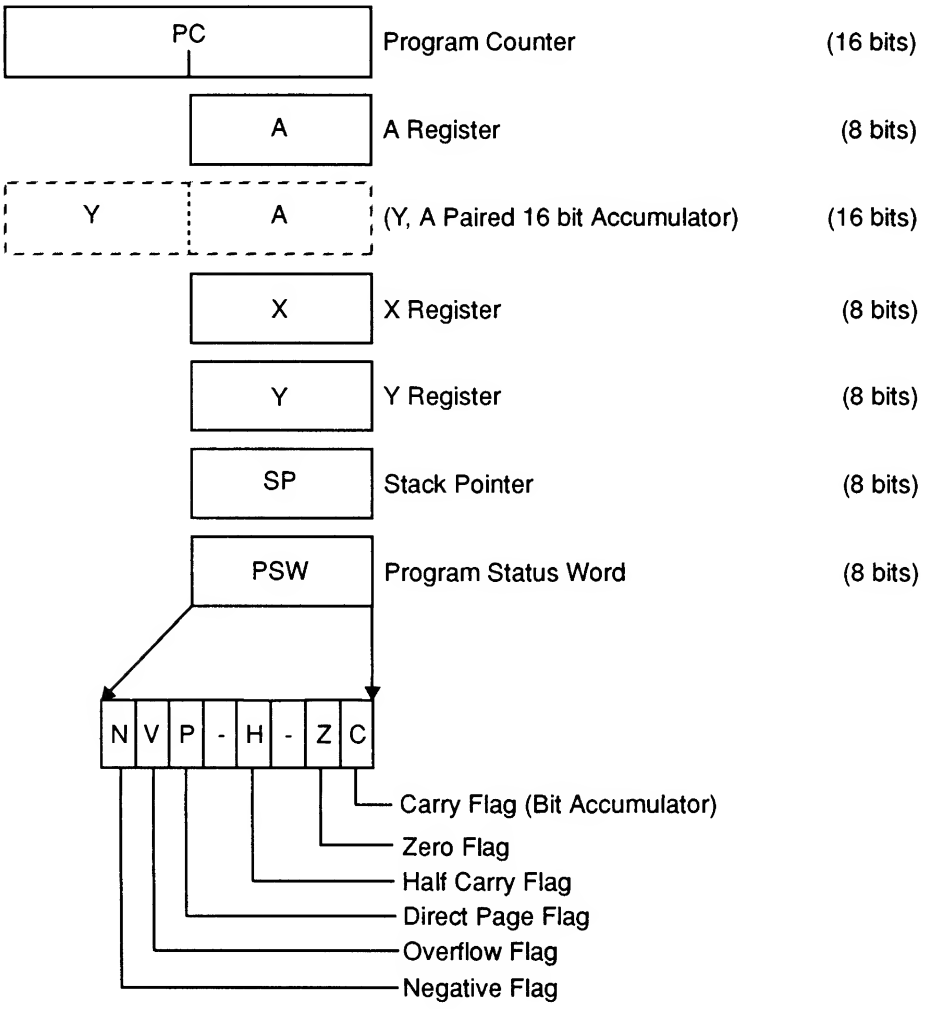


Figure 3-8-1 CPU Registers

8.1.1 A REGISTER

This register is used as an 8-bit accumulator. At times of 16-bit operation commands, it becomes the register which contains low byte data in the 16-bit accumulator, made up of this paired with the Y register. When operation commands are issued, it becomes the multiplier register and low byte data of the product is entered. When division commands are issued, paired with the Y register, it formulates the dividend and the resulting quotient is entered.

8.1.2 X REGISTER

In addition to its role as a universal data register, it also functions as an index register when index addressing is being carried out. In addition, it is used as a two- address command destination address register and X register indirect address register. In division commands, it becomes the divisor register.

8.1.3 Y REGISTER

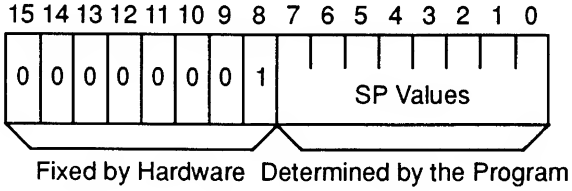
In addition to its role as a universal register, it functions as an index register when index addressing is being carried out. In addition, it is used as a two address command source address register. When carrying out 16-bit operation commands, it becomes the register which contains the high byte data of the 16-bit accumulator, which is made up of the pairing of this register with the A register. When multiplication commands are being carried out, it becomes the dividend register and the product high byte data is entered. When carrying out division commands, paired with the A register it formulates the dividend and the resulting remainder is entered.

8.1.4 PROGRAM COUNTER

The program counter is made up of 16 bits and has an address region of 64 Kbytes. The upper level 8 bits are called PCH and the lower level 8 bits are called PCL. Normally, it will contain the address to be executed during the next machine cycle and will be incremented by only the number of bytes necessary for the command to be fetched. When there is a branching command in the midst of the program, the address of the branch destination will be stored in the program counter. When there is a reset ($\overline{POR}$) input, reset vectors which are in addresses $FFFF_H$ and $FFFE_H$ enter respectively PCH and PCL for branching to take place.

8.1.5 STACK POINTER

The stack pointer is used to send data to the RAM or to recover data from the RAM when the subroutine call commands push (PUSH), pop (POP), or return (RET) are to be carried out. The address region indicated by the stack pointer is within page 1 (addresses 0100_H ~ $01FF_H$).



(NCL PG 28)

When sending data to the RAM, the stack pointer decreases by one after sending data (post decrement) and increases by one prior to restoring data (pre-increment). The diversified activities of the stack pointer are summarized below.

*SUB-ROUTINE CALLS

Stack Address	Activity	SP Value After Sending
SP	Sending to PCH	SP-1
SP-1	Sending to PCL	SP-2

*RESTORING FROM SUB-ROUTINE

Stack Address	Activity	SP Value After Sending
SP	Restore to PCH	SP+1
SP+1	Restore to PCL	SP+2

To send the contents of the A register, X register, Y register, or PSW (program status word) to and from the stack, the commands PUSH and POP can be used.

*PUSH A (X, Y, PSW)

Stack Address	Activity	SP Value After Sending
SP	Sending of A (X, Y, PSW)	SP-1

*POP A (X, Y, PSW)

Stack Address	Activity	SP Value After Sending
SP	Restore A (X, Y, PSW)	SP+1

8.1.6 PROGRAM STATUS WORD (PSW)

The program status word is made up of the various flags which are set and reset according to the results of the execution of 8-bit register commands and the various flags which determine the activities of the CPU. When reset it becomes "000-0-00".

7	6	5	4	3	2	1	0
N	V	P	–	H	–	Z	C

◊ Carry Flag (C)
After operation execution, this flag is set when there has been a carry from the uppermost bit of the arithmetic logic unit (ALU) or when there has been no borrow. It is also altered with shift or rotate commands. It acts as bit accumulator for Boolean bit operation commands. It is set with the SETC command and reset with the CLRC command. The carry flag inverts with the NOTC command.
◊ Zero Flag (Z)
After operation execution, this flag is set when the result is zero and reset when the result is not zero. Even with 16-bit operation commands, zero detection is carried out. It is possible to carry out tests with conditional branching commands.
◊ Half Carry Flag (H)
After operation execution, this flag is set when there has been a carry from bit 3 of the ALU to bit 4 or when there has not been any borrow. There is no command to set the half carry flag however, it is reset by means of the CLRV command. Whenever the half carry flag is set, the overflow flag is also set.
◊ Direct Page Flag (P)
This is the flag which designates the direct page to which many addressing modes are applicable, such as direct page addressing, etc. When "0", the direct page becomes the addresses of the region 0000_H ~ $00FF_H$ and when "1", it becomes the addresses of the region 0100_H ~ $01FF_H$. It is set by the SET P command and reset by the CLRP command.
◊ Overflow Flag (V)
After arithmetic operation execution, this flag is set when overflow or underflow has been produced. When this occurs the H flag is also set. It is possible to carry out tests with conditional branching commands.
◊ Negative Flag (N)
After operation execution, this flag is set when the value of the result of MSB is "1" and reset when its value is "0". It is possible to carry out tests with conditional branching commands.

8.2 MEMORY SPACE

It is possible for the Sound-CPU to address 64 Kbytes of memory. Memory space is divided up according to purpose. From address 0000H, 512 bytes are divided into two pages of 256 byte units called page zero and page one. It is possible to access data within these regions by means of numerous addressing modes, such as direct page addressing, etc. Page one is taken up by the stack.

8.2.1 Direct Pages (Page Zero, Page One)

By means of setting or resetting the Direct Page flag (P) within the program status word, it is possible to designate whether page zero or page one is to be made the direct page. It is set up such that the data within this page can be treated with fewer bytes, at a higher speed, and with more numerous types of commands and addressing modes.

8.2.1.1Stack Area

The stack region is established in the RAM region within page one. The uppermost byte of the stack address is fixed at 01. The lowermost byte of the stack address must be given its initial setting by the program.

8.2.2 Uppermost Page (Internal ROM Region)

A mask ROM is installed within the Sound-CPU from FFC0H - FFFFF. There is a program in it which transmits data from the ROM cassette to the 512 Kbit RAM through the SNES CPU. This region is used by means of reset.

8.2.3 Area of Applicable Bit Operation Commands

8.2.3.1SET1, CLR1

The commands SET1 (set memory bit) and CLR1 (clear memory bit) are applicable to one-bit data with the direct page.

8.2.3.2TSET1, TCLR1

The commands TSET1 (test and set bit) and TCLR1 (test and clear bit) are applicable to the total 64 Kbyte region.

8.2.3.3Boolean Operation Commands

The Boolean operation commands (AND1, OR1, EOR1, MOV1, NOT1) are applicable to the 8 Kbyte region of 0000_H ~ $1FFF_H$.

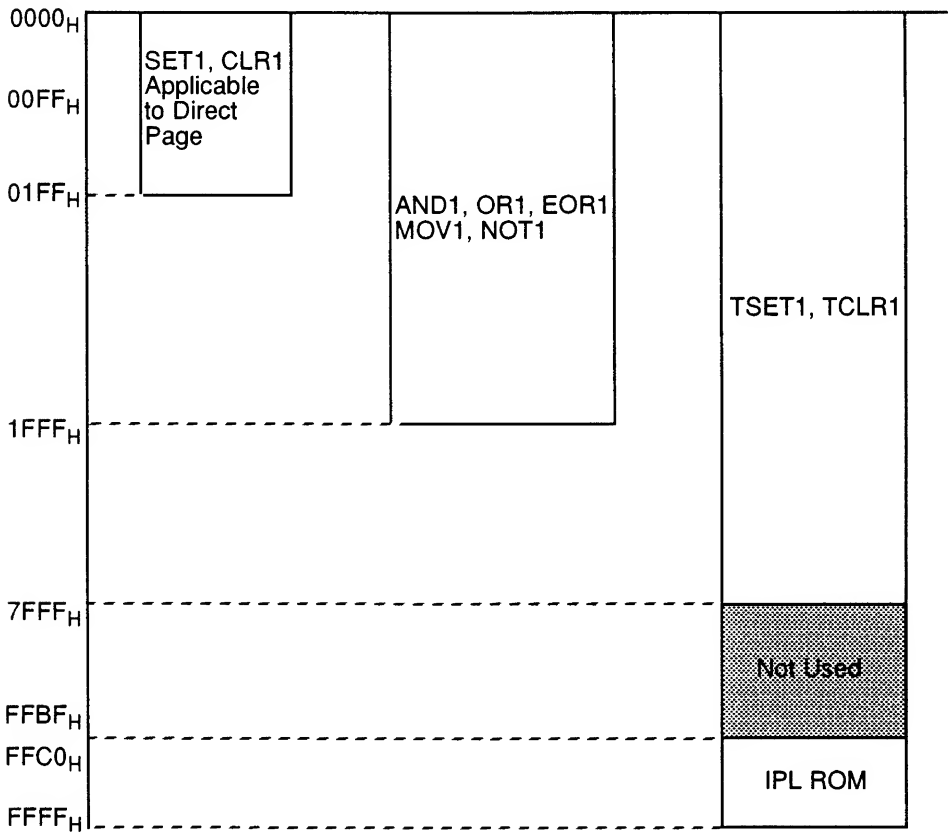


Figure 3-8-2 Boolean Bit Operation Commands

8.2.4 Direct Page Addressing

Since all of the addressing modes indicated in Figure 2-7-3 are applicable to the data of the direct page (P=0: addresses 0000_H ~ $00FF_H$, P=1: addresses 0100_H ~ $01FF_H$) designated by the direct page (P) flag, it is possible to manipulate the data in various ways. In addition, byte efficiency improves due to the fact that direct address designation is possible by one-byte data within the command words. Since effective command cycles also decrease, data can be accessed more rapidly.

Symbol	Addressing	# of Bytes	Effective Address Region		
			0000_H ~ $01FF_H$	~ $1FFF_H$	~ $1FFF_H$
dp	Direct Page	2			
dp+X	X-Indexed Direct Page	2			
dp+Y	Y-Indexed Direct Page	2			
(X)	Indirect	1			
(X)+	Indirect Auto-Increment	1			
dp. dp	Direct Page to D.P.	3			
(X),(Y)	Indirect Page to I.P.	1			
dp,#imm	Immediate Data to D.P.	3			
dp.bit	Direct Page Bit	2			
dp.bit,rel	Direct Page Bit Relative	3			
mem.bit	Absolute Boolean Bit	3			
!abs	Absolute	3			
!abs+X	X-Indexed Absolute	3			
!abs+y	Y-Indexed Absolute	3			
[DP+X]	X-Indexed Indirect	2			
[DP+Y]	Indirect Y-Indexed Indirect	2			

Figure 3-8-3 Memory Access Addressing Effective Address

(NCL PG 33)

Chapter 9. Sound Programming Cautions

9.1 CAUTION #1

When layering sound on several tracks (for example, when layering sound effects on back-ground music), make sure an overflow does not occur due to the additional output. Eight-track sound is ultimately transmitted as one signal, which is limited by the maximum value for the DAC. Distortion noise is created when the signal exceeds this limit.

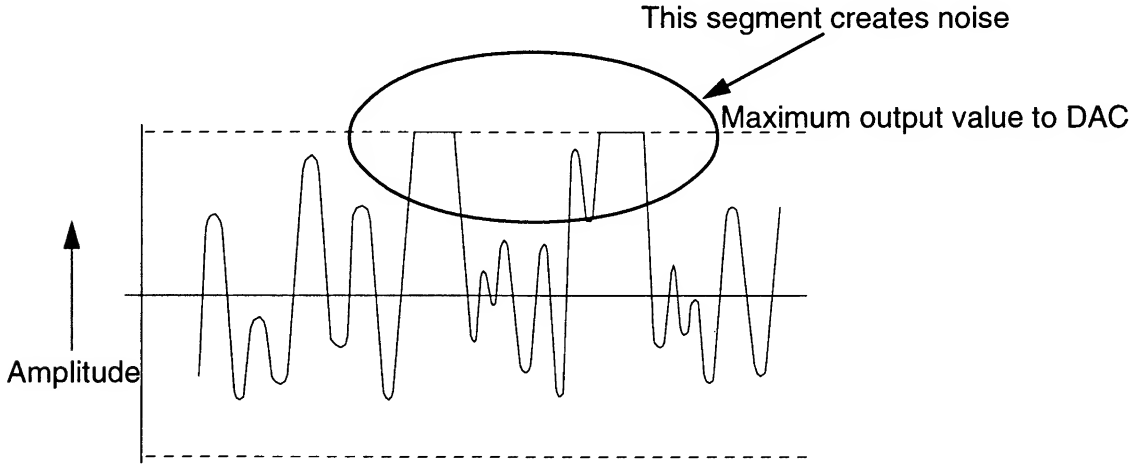


Figure 3-9-1 Wave-form Overflow

9.2 CAUTION #2

The following precautions should be observed when making the initial selections for a sound driver echo function.

1. The FLG's ECEN should not be turned "on" immediately after the EDL and ESA registers have been assigned a number. Otherwise, the RAM area used by the program or other area could be damaged. Either of the following guidelines can be used to determine the appropriate wait period after setting the EDL and ESA registers.

   a) Wait 240 ms.

   b) Read the EDL value (α) before writing to it, and calculate the wait period based on the following formula.

   $$\alpha \times 16 \text{ (ms)}$$

   In addition, the EVOL should be set high only after (the EDL value) x 16 ms or greater. (The read data is undefined until the DSP begins writing data, and could generate noise.)

2. Turn both the ECEN and EVOL "off" when the echo function is not in use. Data will be read and output unless the EVOL is 0.

9.3 CAUTION #3 (ECHO OPERATIONS)

This caution describes the procedure to be followed when writing echo data to the appropriate RAM area.

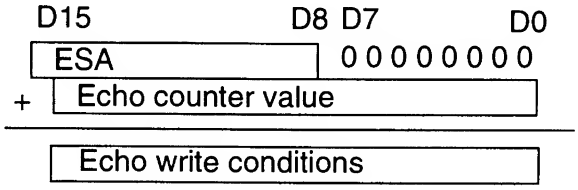
- ESA (Echo Start Address - 6DH): Initial address for the echo start area.
- EDL (Echo Delay - 7DH): Determines the number of addresses in the echo area begining from the initial address.

9.3.1 PROCEDURE

An internal counter exists for the echo data, which is written sequentially. This counter is called the "echo counter". The EDL determines the maximum value of the echo counter. When the echo counter reaches (the ESA value x 80H), the echo counter is set to 00. Echo data is written two bytes at a time (4 bytes for the left and right) every 31.25 s. A delay of 16 ms occurs using a RAM address area of 80H.

The echo counter is 15 bits, from 000H ~ 7FFH. The following formula is used to determine the RAM address to which the echo data is written.

(ESA value x 100H) + (echo counter value) = (echo write address)



However, changes in the echo counter value do not immediately follow changes in the ESA value using the above formula. Therefore, unanticipated problems, such as data loss, could occur. The relationship between the echo counter and EDL value can be explained as follows.

It was mentioned above that the echo counter is set to 00 when it reaches (the ESA value x 80H). However, the ESA value mentioned here is not the value of ESA at that time, but at the time when the echo counter was previously set to 00. Even when the ESA value is changed, the counter continues counting until it reaches the previously set ESA value, wherein it is set to 00. Then, the last-specified ESA value and echo counter value are compared. (This is to prevent the echo counter from incrementing until the maximum value is reached, when a small value is assigned to ESA). For these reasons, the echo write address will be within the specified range if the programmer waits for the period of time specified below when re-writing the EDL value.

wait time = (the EDL value prior to rewrite) x 16 ms

To insure that the echo data is written to the echo area, wait for a period equal to the last-specified EDL value x 16 ms.

9.4 CAUTION #4

Always select appropriate values for the echo parameters. Inappropriate values could lead to loss of data in critical RAM areas or noise generation. Reverberation may occur when the echo feedback value is too large.

9.5 CAUTION #5

It is extremely important to follow the recommended procedure (Caution #3, above) when setting the initial echo values and modifying the echo parameters. The RAM used as the echo buffer is also used for the program, wave form data, and sound driver. If an echo is started before the echo parameter initialization is established, critical data may be overwritten in the RAM area.

9.6 CAUTION #6

Do not use an excessive sound data compression ratio. An excessive compression ratio results in distorted sound output.

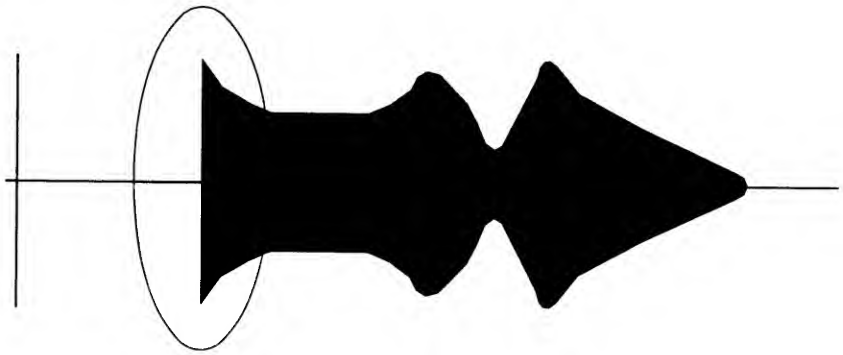
9.7 CAUTION #7

When performing sound checks, the monaural sound output should also be checked. Sound data created for stereo output may not be produced as desired when played on a monaural output device. Super NES monaural sound is generated by adding stereo sound output in the circuit. When, for example, a phase effect is created in stereo by setting negative values in the volume register, the sound volume may be altered when the sound is combined to generate monaural output.

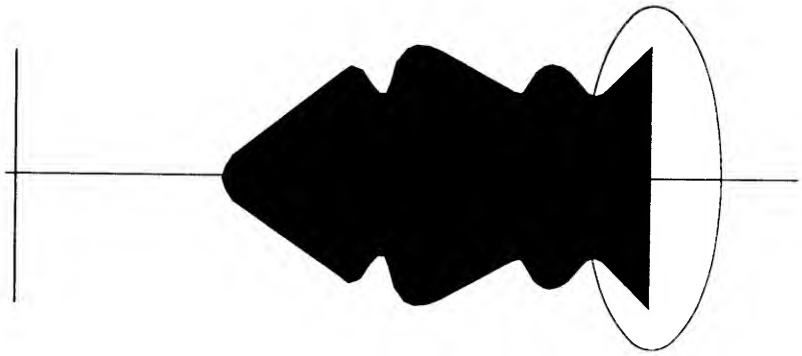
9.8 CAUTION #8

Sampled data should not have any discontinuity. A crackling noise is produced by discontinuous samples. The following are examples of discontinuity in the sampled data.

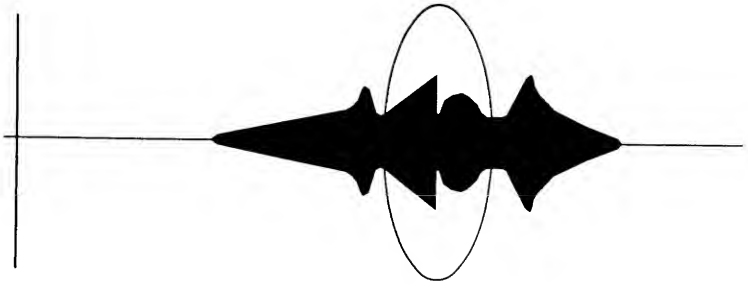
- The sampled data does not begin at 0.



- The sampled data does not end at 0.



- A discontinuity occurs in the middle of the sampled data.



9.9 CAUTION #9

When transferring data between the Super NES CPU and the APU using the IPL loader, a hang-up can occur if the program is interrupted.

When the Super NES CPU sends the termination code, the Sound CPU sends a code to the Super NES CPU to indicate that it has received data. The Sound CPU erases this code after 300-400 µsec.

If an interrupt occurs after sending the termination code, for a period which is greater than 300-400 micro-seconds, the status code from the Sound CPU will be erased before it can be read by the Super NES CPU. The hang-up will occur because the Super NES CPU will wait indefinitely for the Sound CPU to indicate that it has received data.

Two possible options are available to prevent this from occuring.

- Modify the transfer routine run on the Super NES CPU side.
- Inhibit interrupts during transfer.

These options are demonstrated below.

9.7.1 MODIFIED TRANSFER ROUTINE

Add two lines as shown to the routine.

```
                .
                .
                .
                adc   #07fh              ; original code
                pla                      ; original code
                sta   !APU_port 0        ; original code
                cpx   #1                 ; solution #1
                bcc   boot_ret           ; solution #1
boot_wait3      cmp   !APU_port0         ; original code
                bne   boot_wait3         ; original code
                bvs                      ; original code
boot_ret        plp                      ; original code
                rts                      ; original code
                end
```

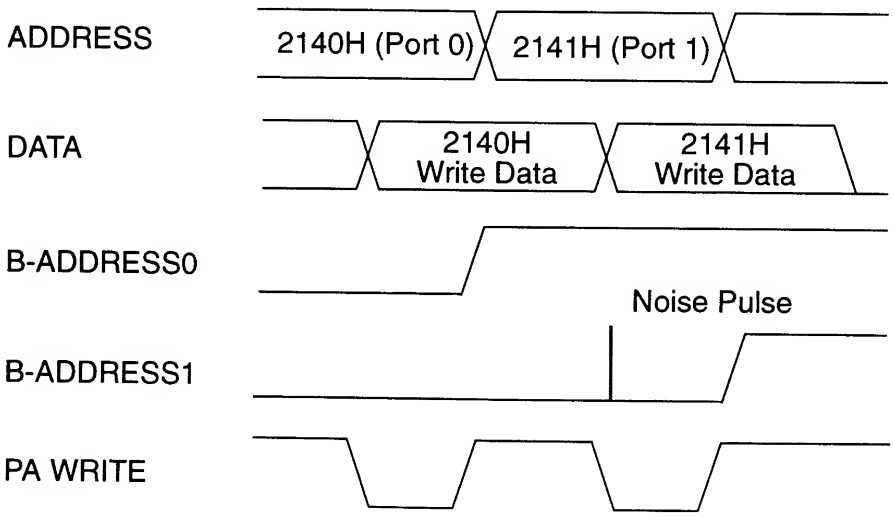
9.7.2 INHIBITING INTERRUPTS

Inhibit any interrupt from the time the termination code is sent until the Sound CPU sends an acknowledgement. This is demonstrated by the highlighted code below.

```
                .
                .
                .
                adc   #07fh                ; original code
                pla                        ; original code
                sta   !APU_port 0          ; no interrupt
boot_wait3      cmp   !APU_port0           ; no interrupt
                bne   boot_wait3           ; no interrupt
                bvs                        ; original code
                pop                        ; original code
                rts                        ; original code
                end
```

9.10 CAUTION #10 - DATA TRANSFER

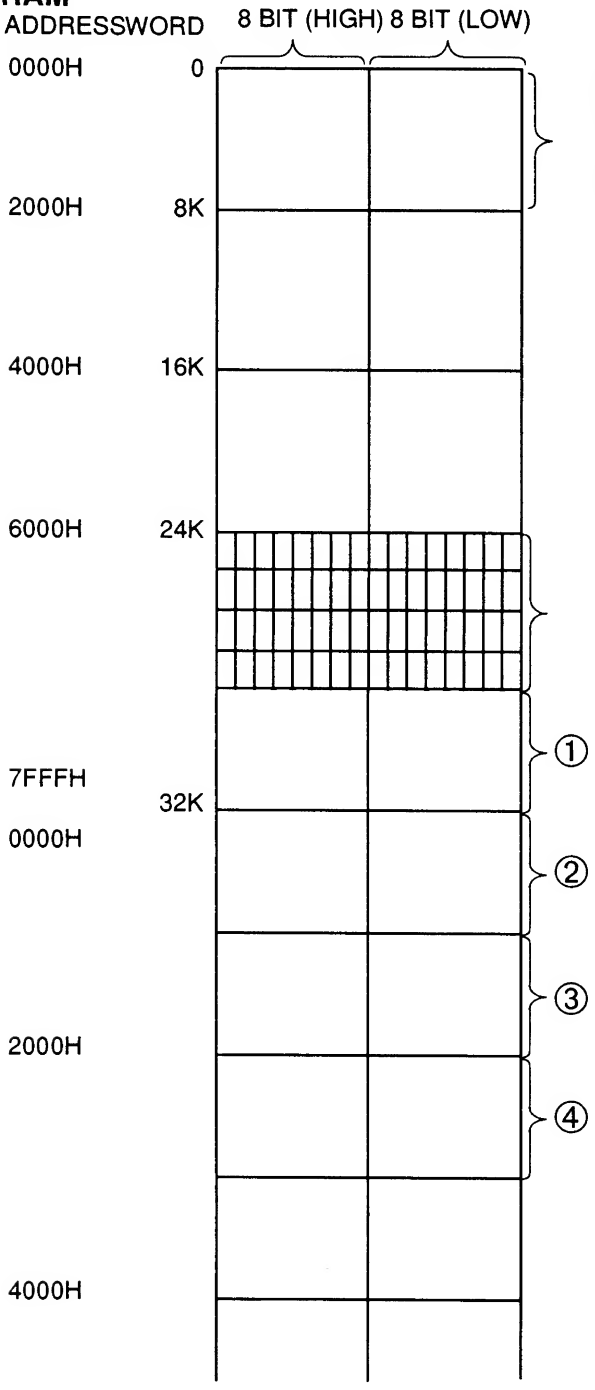
When data is written to Port 0 <2140H> and Port 1 <2141H> in the 16 bit mode, during data transfer from the Super NES APU, the value of Port 3 <2143H> may, inadvertantly, be changed. Therefore, the 8 bit mode should be used when writing data to these ports.
This occurs because multiple ROMs installed on the game pak PCB can increase load capacity of the data bus and, when combined with a drastic fluctuation of CPU data output, cause noise in the data being written. An example is provided below.



In the example on the previous page, if data is written to 2140H (Port 0) and 2141H (Port 1) in the 16 bit mode, noise pulses may occur at B-Address 1 due to noise which occurs when all CPU data simultaneously changes from high to low. This depends upon the type of CPU data. When data is written to 2141H (Port 1), B-Address1 becomes "1". In other words, the same data is written to 2143H (Port 3) due to this pulse.

Appendix A. PPU Registers

V-RAM



8K-WORD: This is an area which is designated by "OBJ NAME BASE ADDRESS" of the register <2101H>. (32K-WORD / 4-Partition.) [The BA2 of the register <2101H> "OBJ NAME BASE ADDRESS" is used for expansion purposes, and it will normally be ignored.]

[In case BA1=1 and BA0=1 are set by "OBJ NAME BASE ADDRESS"]

4K-WORD: This is a lower 4K-WORD of the area (8K-WORD) designated by "OBJ NAME BASE ADDRESS" of the register <2101H>. The combination of this 4K-WORD and the 4K-WORD remaining will be determined by "OBJ NAME SELECT" of the register <2101H>.

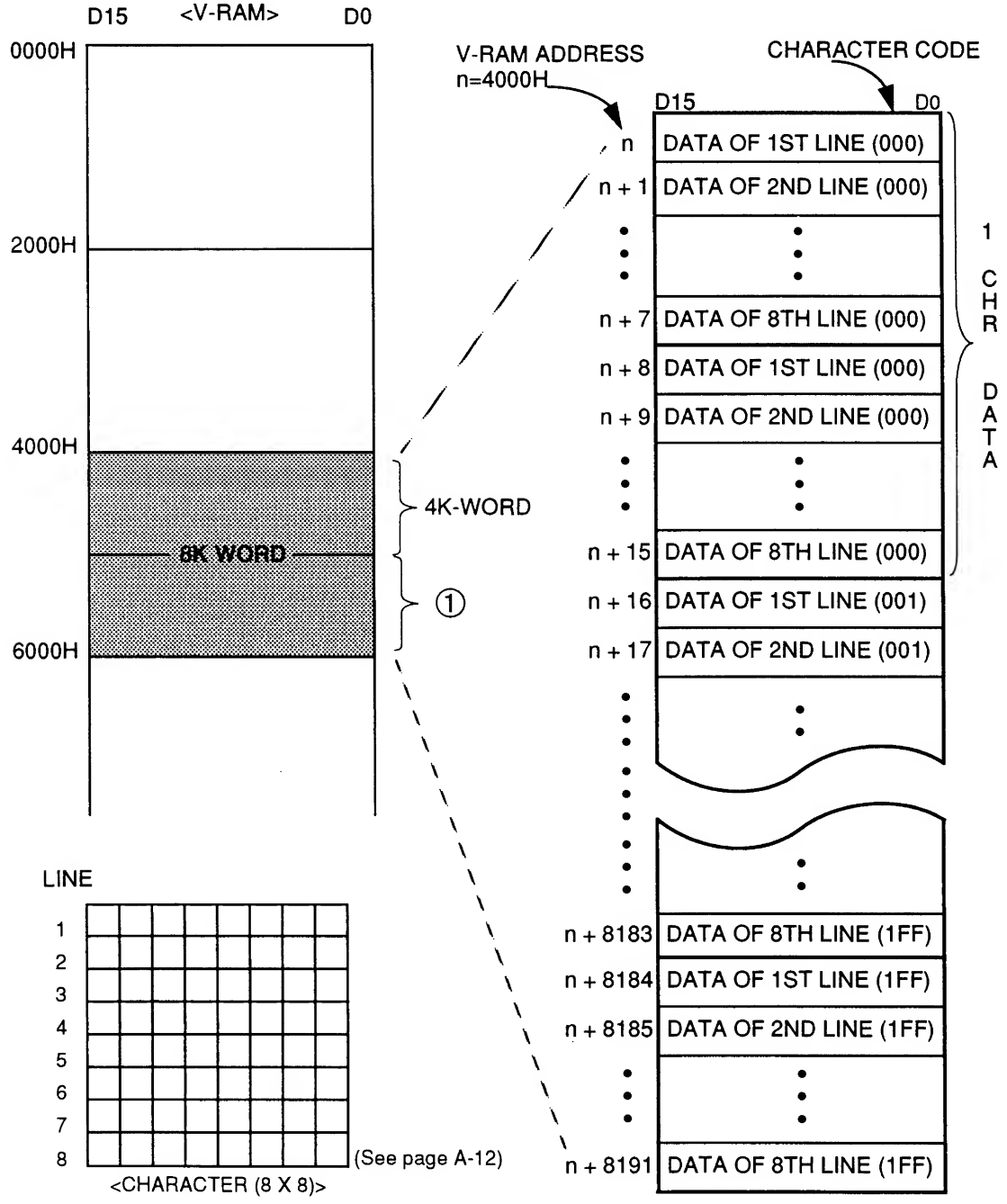
OBJ Name Select

N1	N0	COMBINATION
0	0	4K - WORD + ①
0	1	4K - WORD + ②
1	0	4K - WORD + ③
1	1	4K - WORD + ④

(NCL PG 68)

OBJECT DATA TO BE STORED

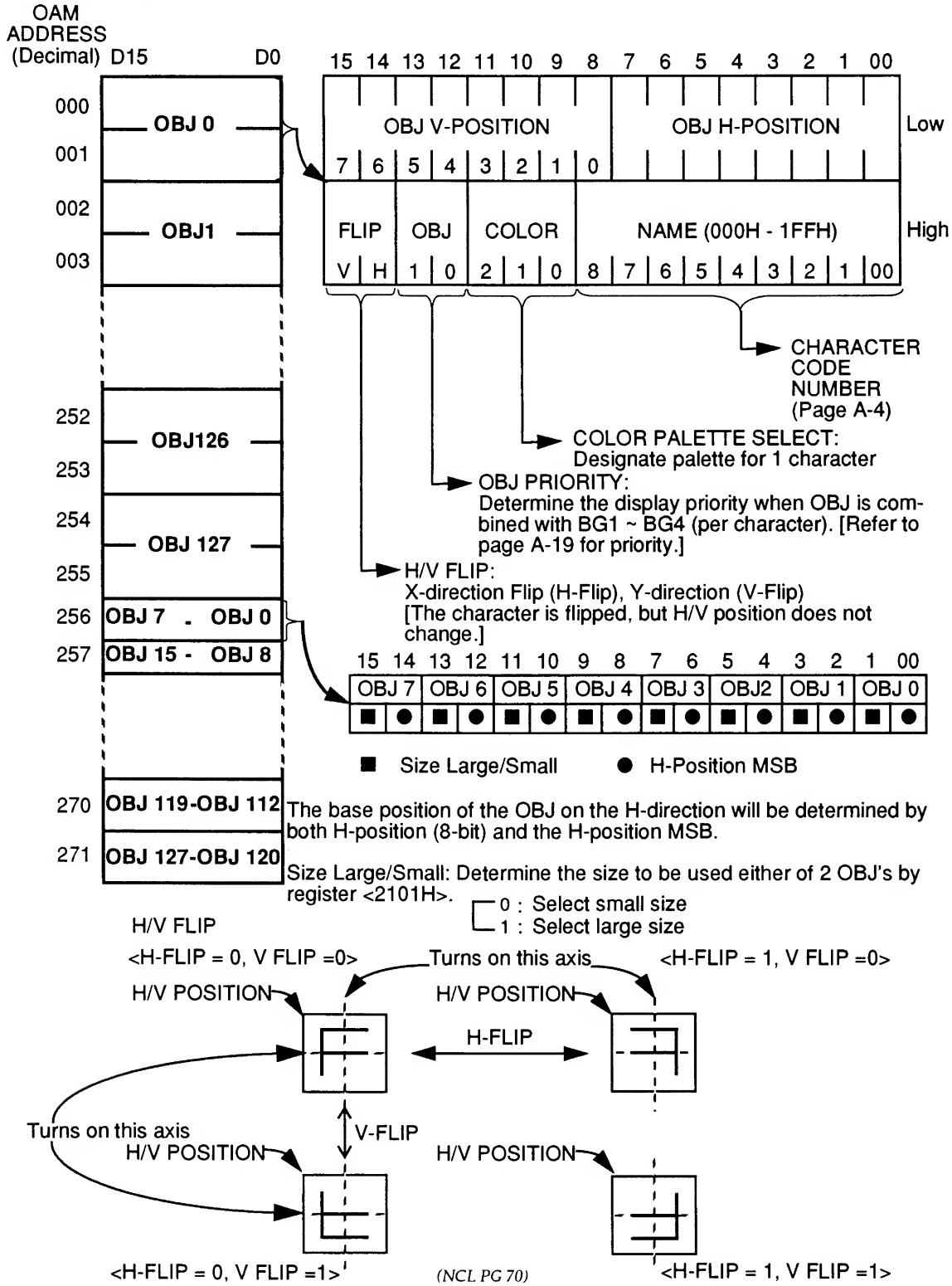
4 BIT CONSTRUCTION [8 x 8 x 4 Bit (16 WORD) / CHARACTER] (Refer to page A-12)
8 x 8 (Character Size) x 4 (Bit Construction) x 512 (Number of character) ⟶ 16K-BYTE
[In case BA1=1 and BA0=0 are set by "OBJ NAME BASE ADDRESS" and also N1=0 and N0=0 are set by "OBJ NAME SELECT"]



(See page A-12)

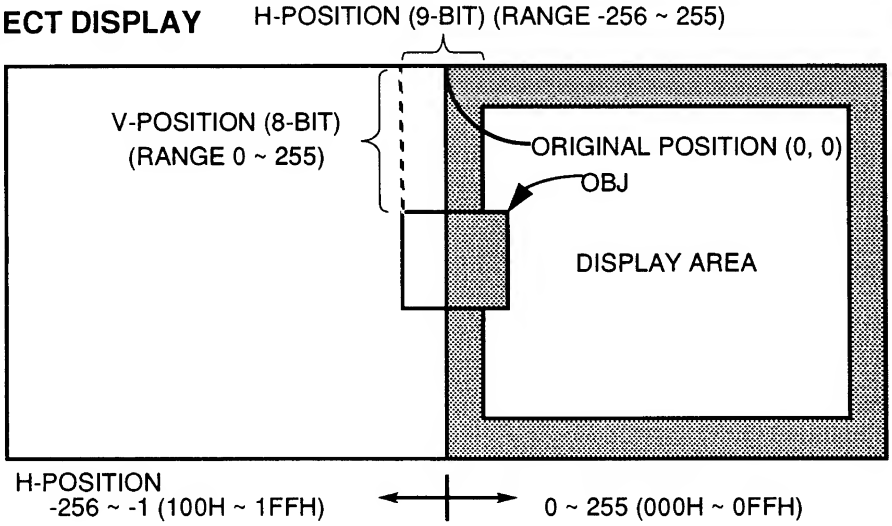
(NCL PG 69)

OBJECT DATA



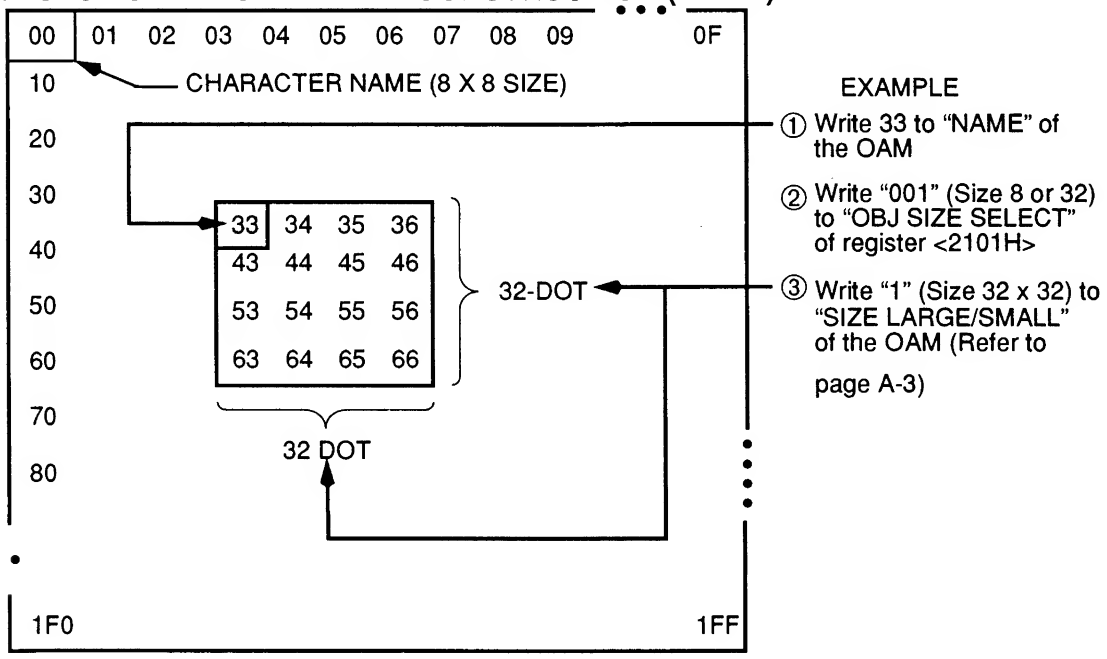
(NCL PG 70)

OBJECT DISPLAY



(NOTE-1) The H-position is a complementary expression of 2 (9-bit).
(NOTE-2) The coordinate of the OBJ displayed is shifted down compared to the coordinate of the BG displayed. [Interlace: 2-dot / Non-Interlace: 1-dot] (See page A-10.)
(NOTE-3) "100H" is basically prohibited to use for 9-bit of the H-Position. (If it is used, it must be counted as OBJ quantity displayed even if it is not displayed on the screen.)

OBJECT CHARACTER DATA CONSTRUCTION (VRAM)



In case the character code is 000 through 0FF, the V-RAM address per character data (16-word) will be "n (Name Base Address) + N (Name) x 16 ~ n + N x 16 + 15." If the character code is 100 through 1FF, it will be "n + Ns (Name Select) x 4K + N x 16 ~ n + Ns x 4K + N x 16 + 15."

(NCL PG 71)

OBJECT

# OF CELLS DISPLAYED	1 2 8			
CELL SIZE	8X8	16X16	32X32	64X64
# OF LINES DISPLAYED	32-pcs (converted to 8x8 size)			
# OF CELL-COLOR	1 6			
# OF PALETTE	8			
# OF COLOR ON SCREEN	1 2 8			
ATTRIBUTE	H-FLIP, V-FLIP FUNCTION DISPLAY PRIORITY (Select priority against BG)			

BG

MODE	# OF SCREENS DIS-PLAYED	SCREEN	# OF CELL DOT	# OF CELL COLOR	# OF PALETTES	# OF COLORS PER SCREEN	FUNCTION
0	MAX 4	BG1	8 X 8	4	8	32	①②③ ⑤⑥⑦⑧ ⑩
		BG2	OR	4	8	32	①②③ ⑤⑥⑦⑧ ⑩
		BG3		4	8	32	①②③ ⑤⑥⑦⑧ ⑩
		BG4	16 X 16	4	8	32	①②③ ⑤⑥⑦⑧ ⑩
1	MAX 3	BG1		16	8	128	①②③ ⑤⑥⑦⑧ ⑩
		BG2		16	8	128	①②③ ⑤⑥⑦⑧ ⑩
		BG3		4	8	32	①②③ ⑤⑥⑦⑧ ⑩
2	MAX 2	BG1		16	8	128	①②③ ⑤⑥⑦⑧ ⑩⑪
		BG2		16	8	128	①②③ ⑤⑥⑦⑧ ⑩⑪
3	MAX 2	BG1		256	1	256	①②③ ⑤⑥⑦⑧⑨⑩
		BG2		16	8	128	①②③ ⑤⑥⑦⑧ ⑩
4	MAX 2	BG1		256	1	256	①②③ ⑤⑥⑦⑧⑨⑩⑪
		BG2		4	8	32	①②③ ⑤⑥⑦⑧ ⑩⑪
5	MAX 2	BG1	▼	16	8	128	①②③ ⑤ ⑦⑧ ⑫
		BG2		4	8	32	①②③ ⑤ ⑦⑧ ⑫
6	1	BG1	16X8	16	8	128	①②③ ⑤ ⑦⑧ ⑪⑫
7	1	BG1	8 X 8	256	1	256	① ③④⑤⑥⑦⑧⑨⑩
EXT BG	1	BG2	8 X 8	128	1	128	① ③④⑤⑥⑦⑧⑨⑩

(NCL PG 72)

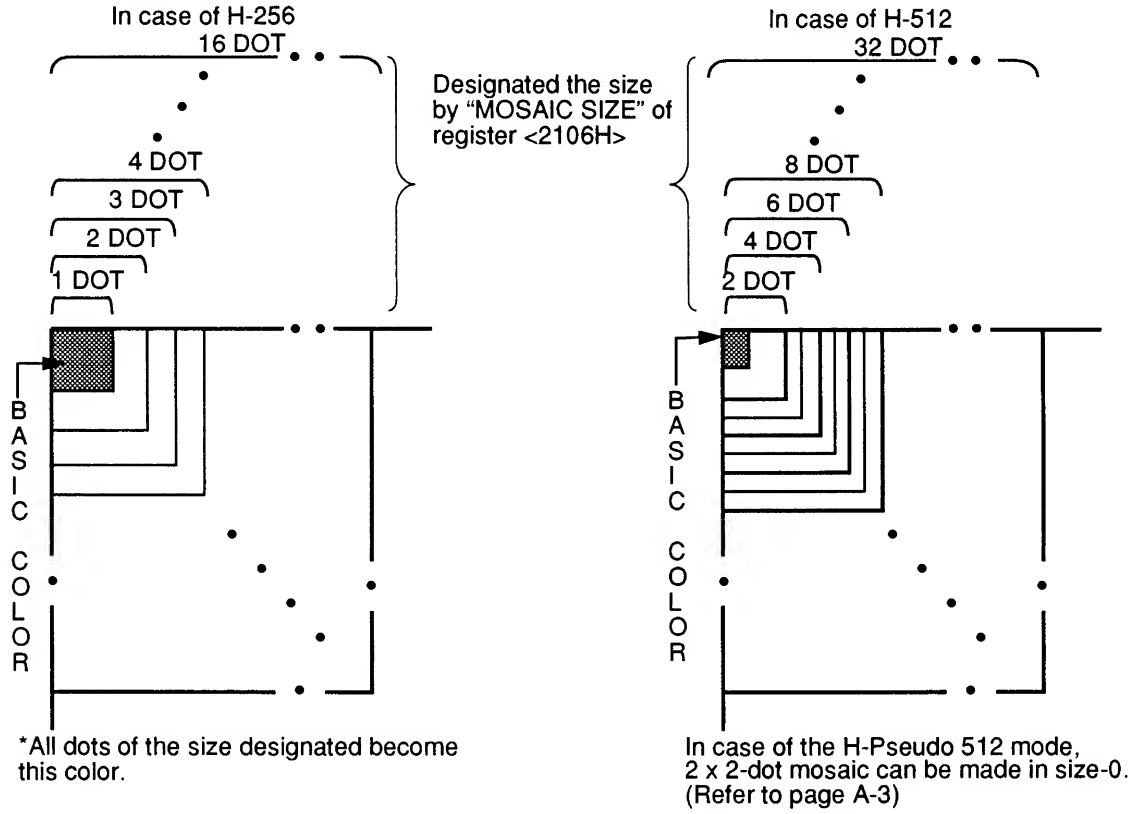
[Main Function of BG]

1. HV Scroll (each screen)
2. HV Flip (each character)
3. Mosaic (Refer to Chapter 4)
4. Rotate, Enlarge, Reduce (Refer to Chapter 5)
5. Window Mask (Refer to Chapter 6)
6. Screen Addition and Subtraction (Refer to ¶7.1)
7. Fixed Color Addition and Subtraction (Refer to ¶7.2)
8. Color Window (Refer to ¶7.2)
9. CG Direct Select (Refer to Chapter 8)
10. Horizontal Pseudo 512 (Refer to Chapter 9)
11. Offset Change (Refer to Chapter 12)
12. Horizontal 512 Mode (Refer toChapter 19)

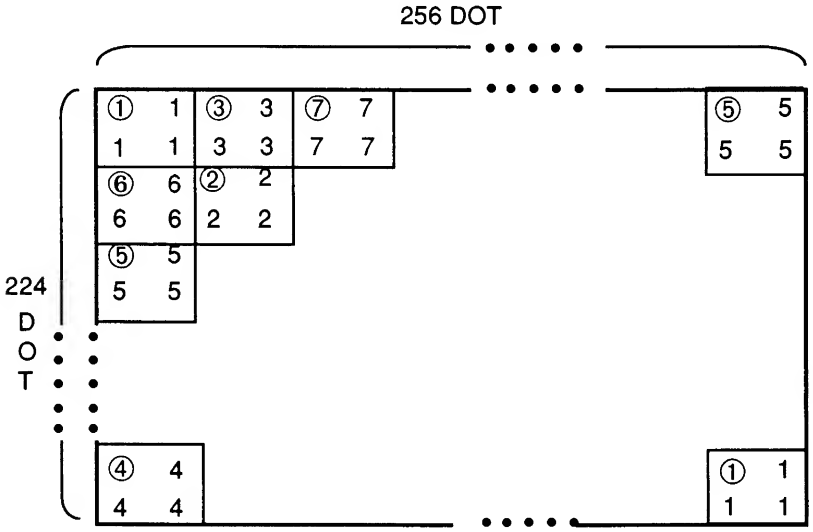
[Other Function]

- Priority (each character/mode 0 ~ 6)
- Screen HV Rotate (mode 7)

MOSAIC SCREEN



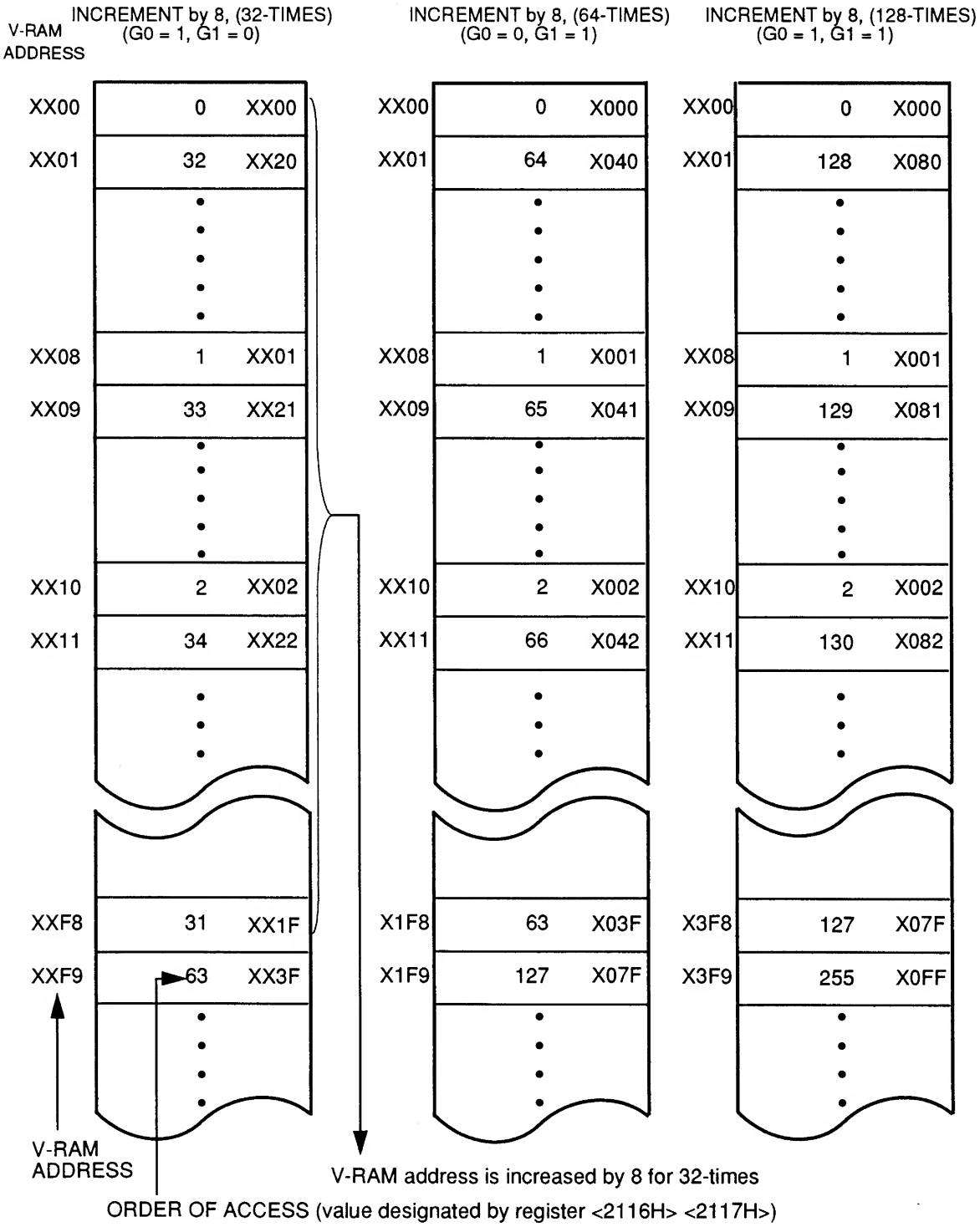
MOSAIC SCREEN DISPLAY EXAMPLE (BG SCREEN)
(When the mosaic size is 2 x 2-dot in the 256-mode)



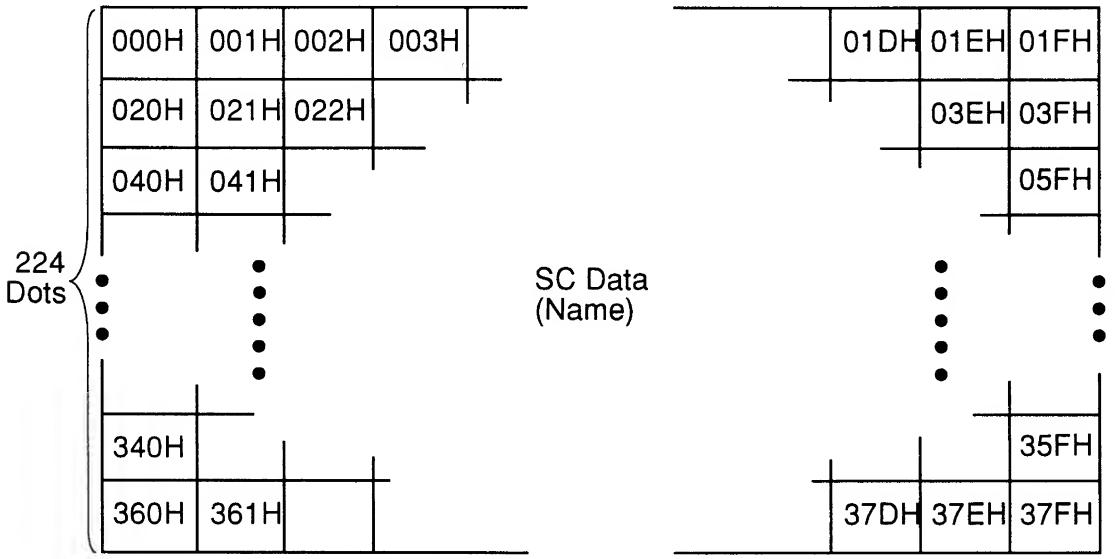
⑪ is the basic color data

(NCL PG 73)

ADDRESS INCREMENT ORDER

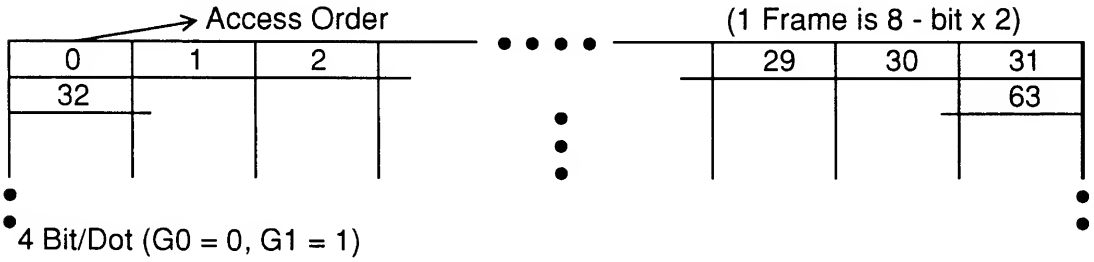


(NCL PG 74)

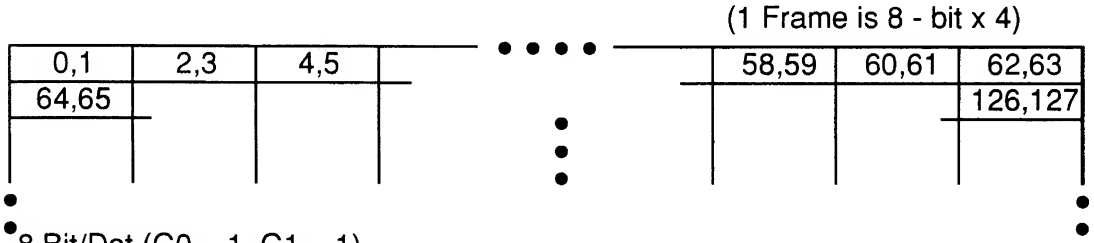


When SC data (Name) is set during BG Mode 0 ~ 6 as demonstrated in the table above, Character data accesses horizontally by 8 dots in Full Graphic (G0, G1) of register <2115H>.

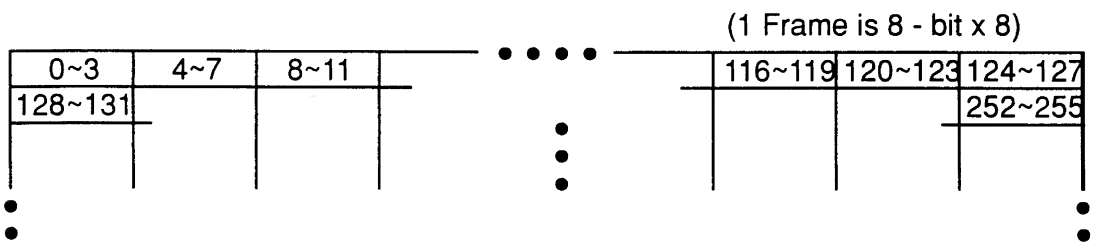
2 Bit/Dot (G0 = 1, G1 = 0)



4 Bit/Dot (G0 = 0, G1 = 1)

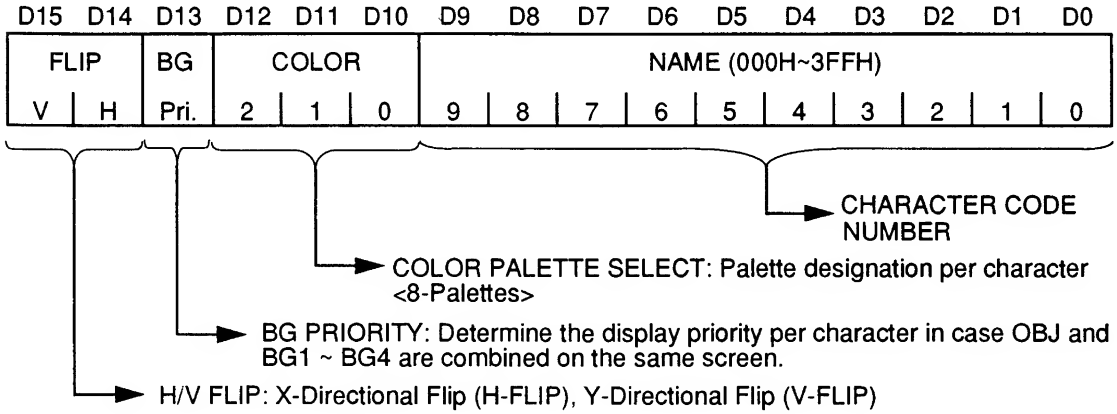


8 Bit/Dot (G0 = 1, G1 = 1)



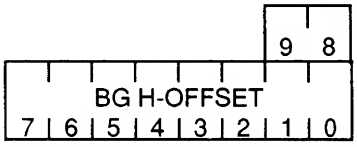
(NCL PG 74a)

BG SC DATA (MODE 0 ~ 6)

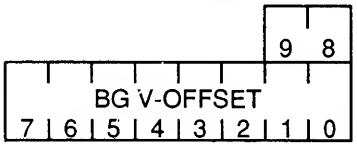


BG SCREEN H/V SCROLL

REGISTER <210DH> <210FH>
<2111H> <2113H>



REGISTER <210EH> <2110H>
<2112H> <2114H>



RANGE OF H-SCROLL
0 ~ 1023 DOT

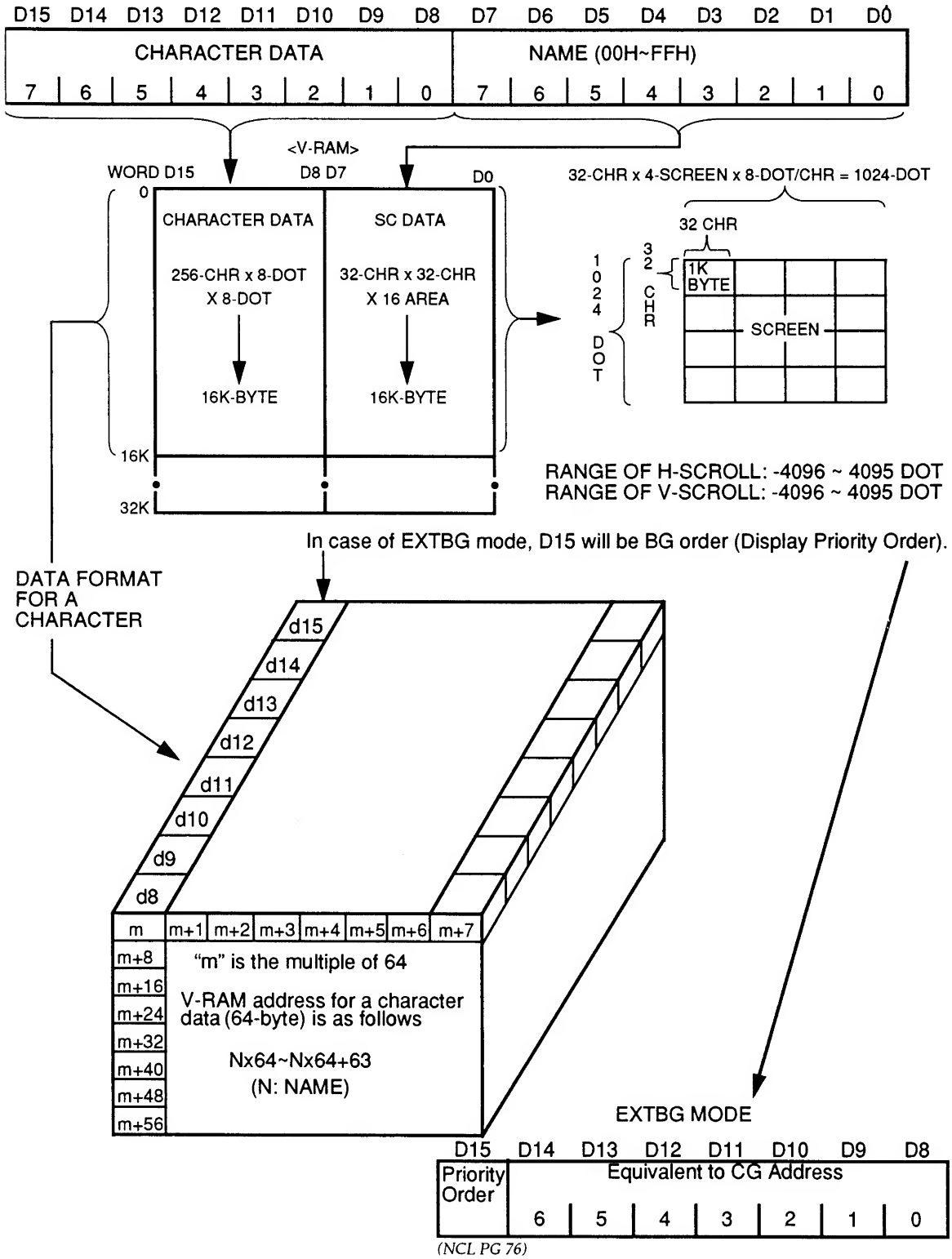
RANGE OF V-SCROLL
0 ~ 1023 DOT

H/Vscroll range may be changed and 2-dot scroll may be possible, depending on the combination of the modes (512, 16-size, interlace, etc.). Also, SC size may be changed against the screen. (Page A-21 and A-22)

$\overline{V}$ coordinate of BG and OBJ has one - line gap.

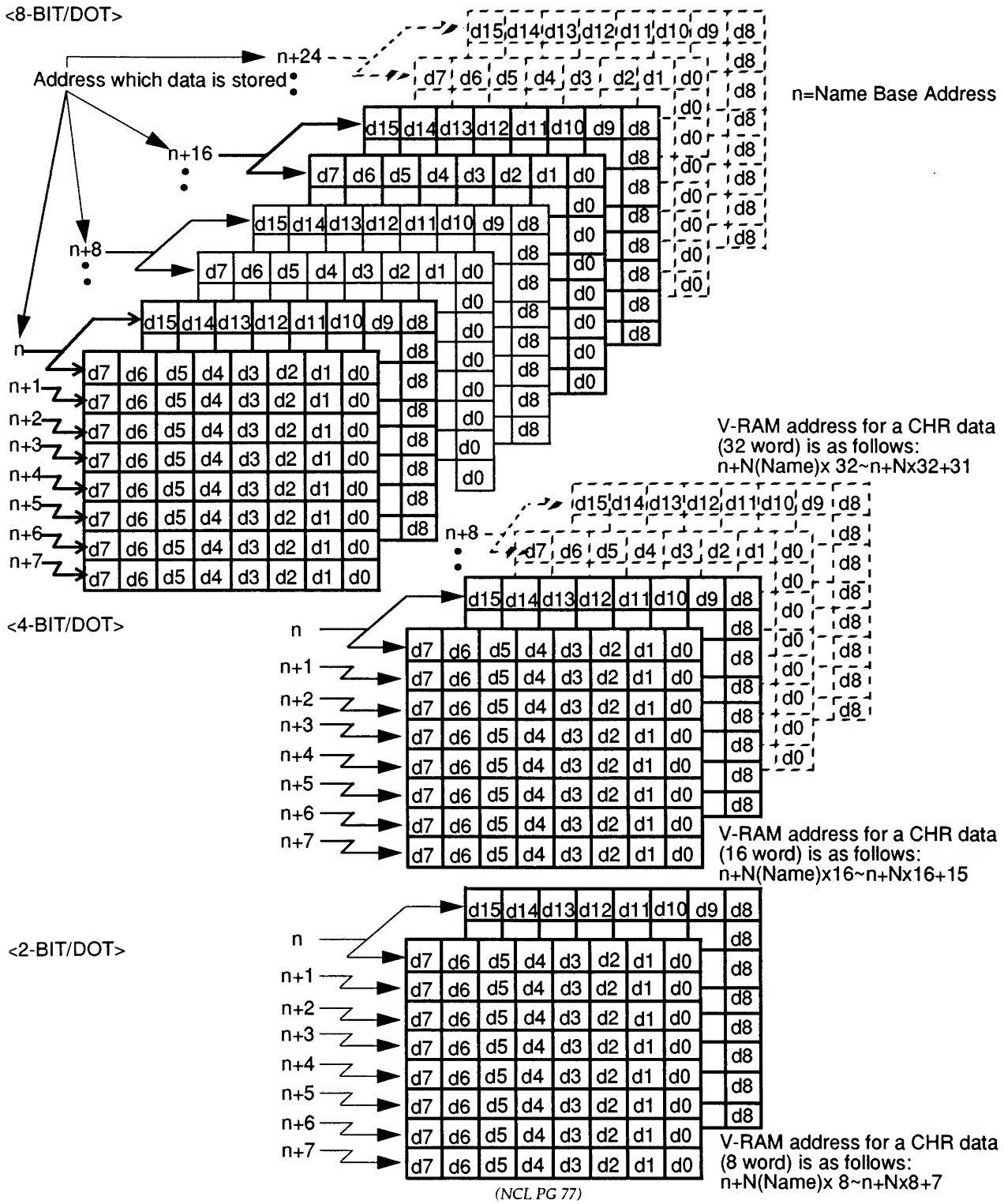
(NCL PG 75)

BG SC DATA (MODE 7)



(NCL PG 76)

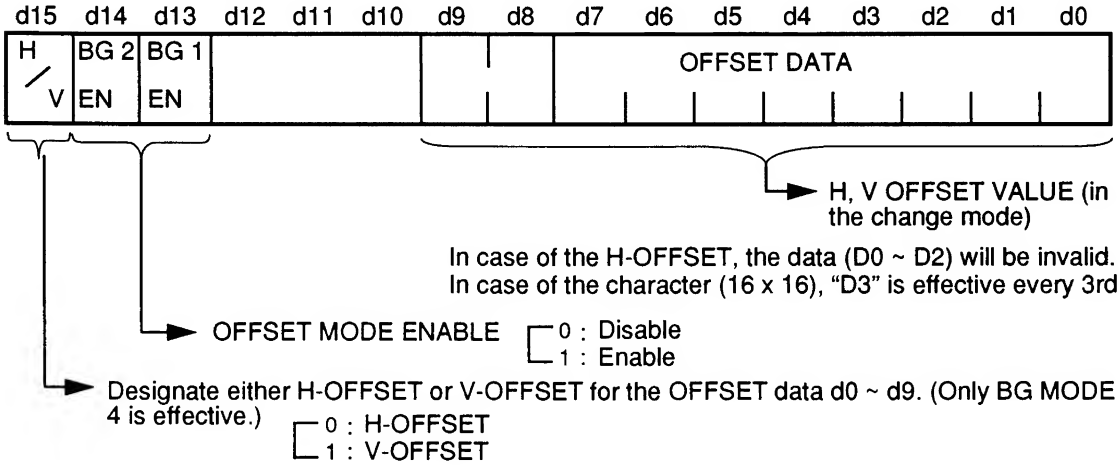
CHR DATA CONSTRUCTION



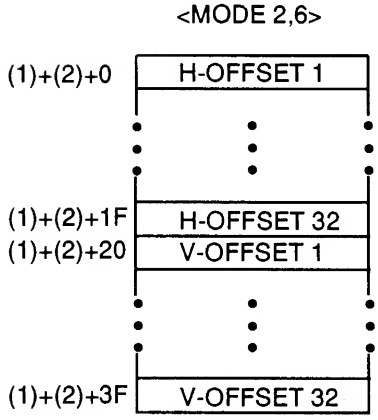
(NCL PG 77)

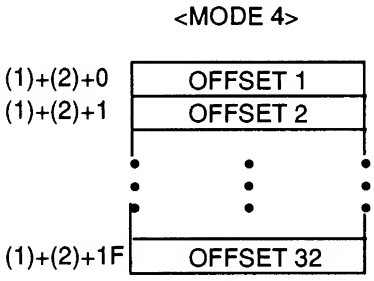
OFFSET CHANGE MODE

The offset change mode can be used in the BG mode 2, 4 and 6, and the following data is required in this mode.



Write this data to VRAM of address designated at (1) and (2), using the BG Mode. (See below.)

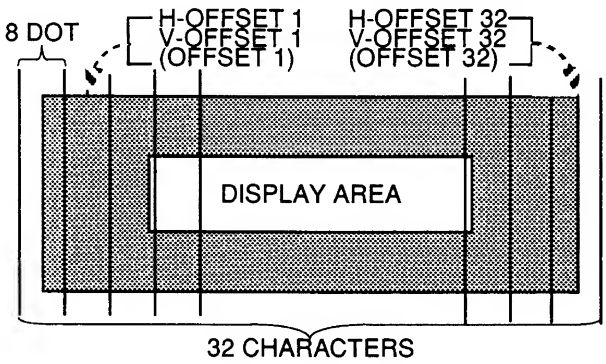




In case BG3 SC size is S1=0, S0=0

(1): BG3 SC Base Address ([value set by "d2" ~ "d7" of <2109H>]x1024)

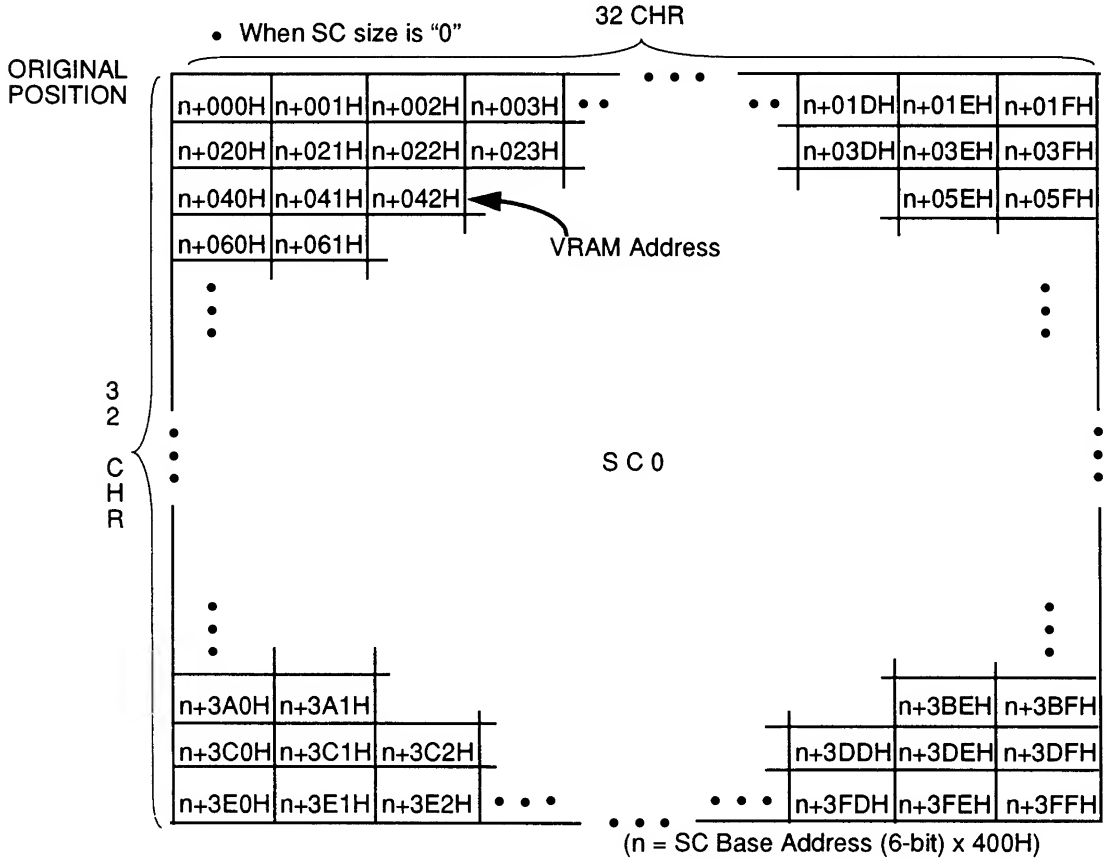
(2): BG3 SC Offset Address ([value set by "d3" ~ "d7" of <2112H>]x32) + ([value set by "d3" ~ "d7" of <2111H>])

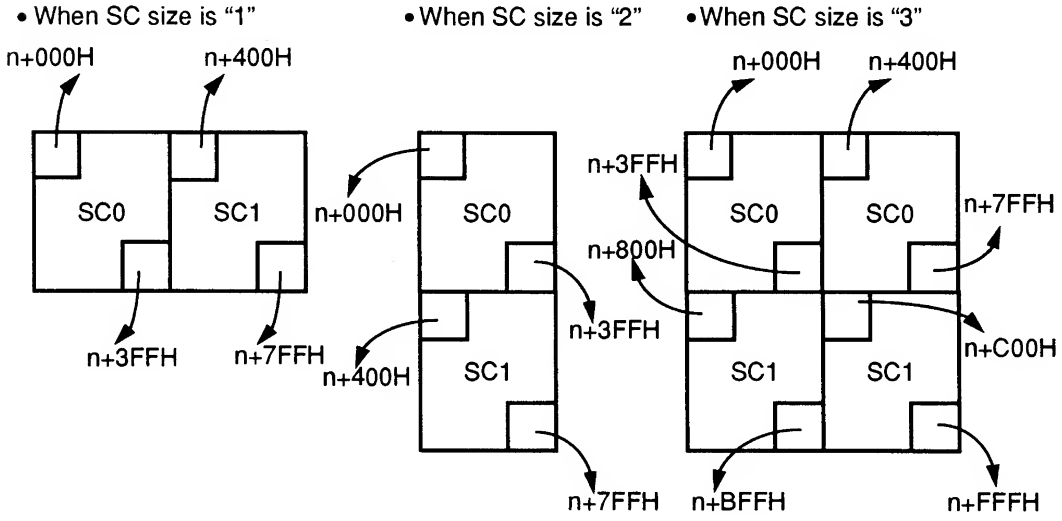


The offset value can be changed by each column (character unit).
(Up to 3rd character can be seen horizontally on the screen by setting the offset value of the entire screen, but the offset can not be changed for 1st character (0 character).

(NCL PG 78)

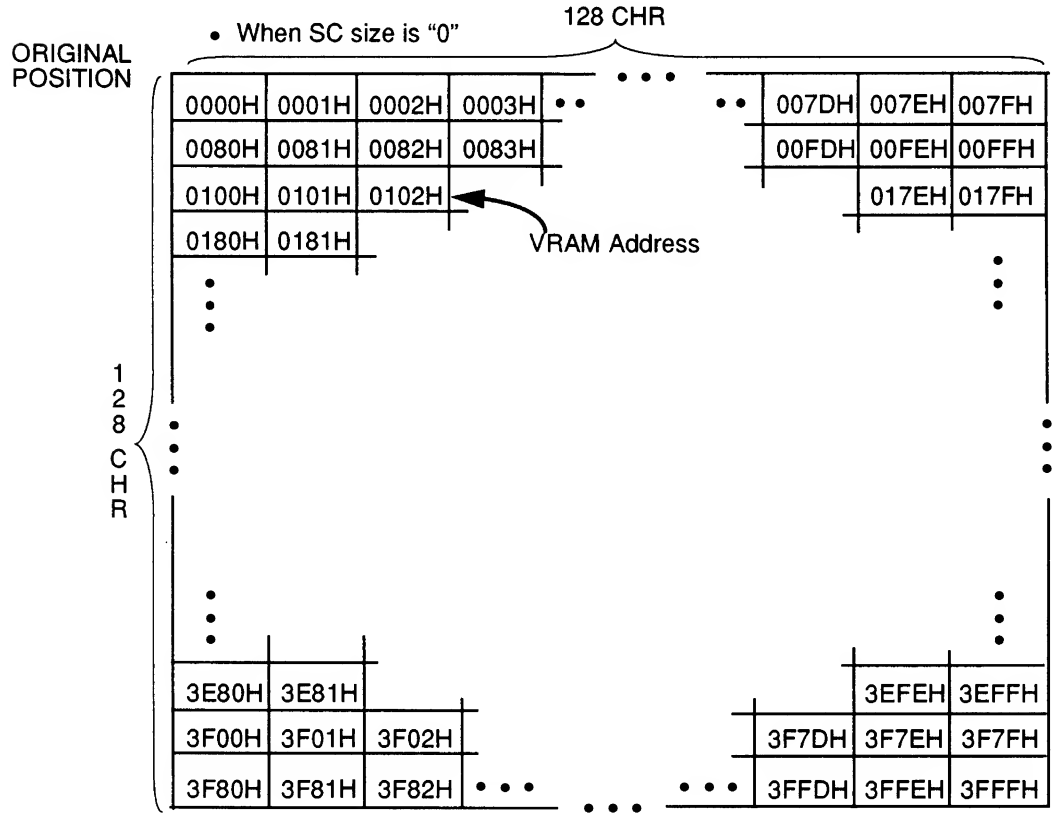
BG SCREEN (BG Mode 0 ~ 6)





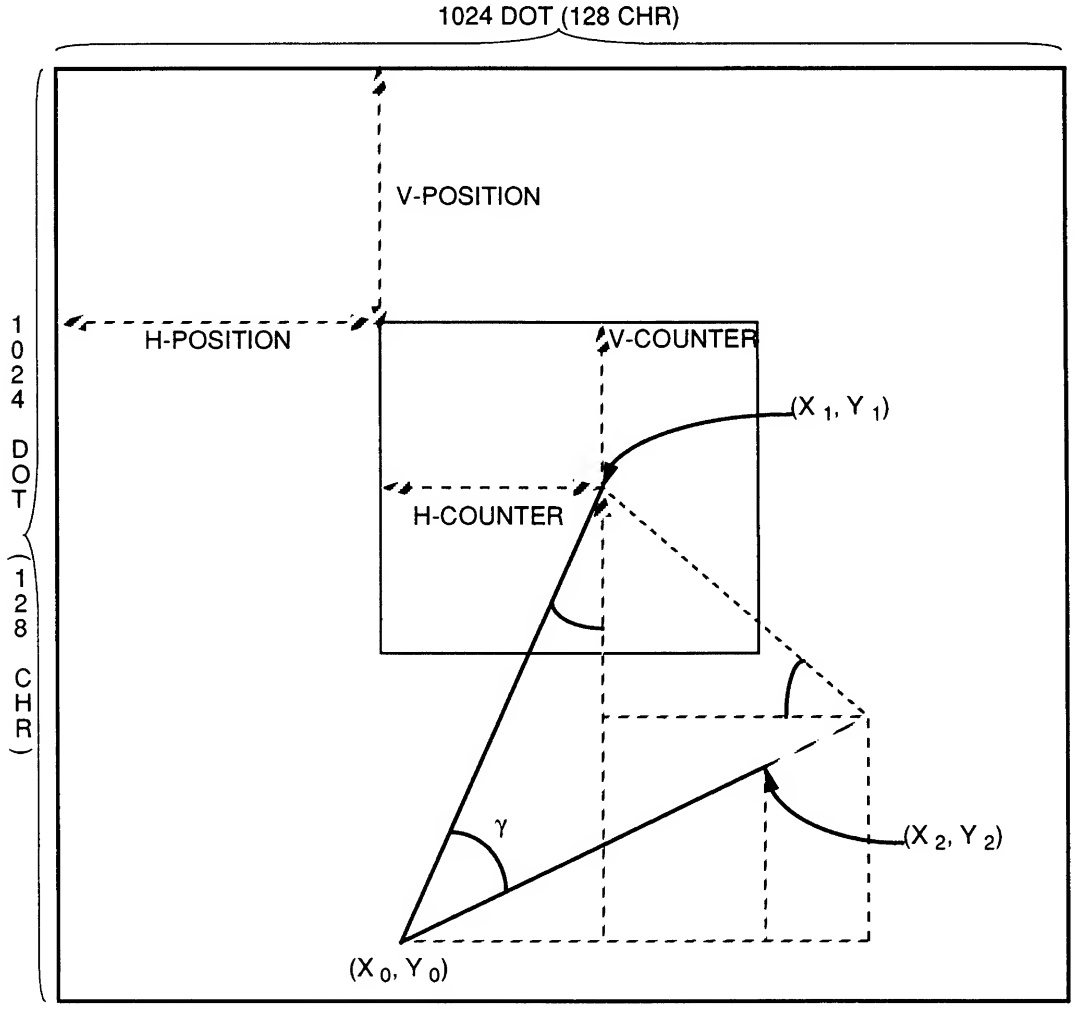
(NCL PG 79)

BG Screen (BG Mode 7)
Screen Size and Area are Fixed



(NCL PG 79a)

OPERATION (ROTATION/ENLARGEMENT/REDUCTION)



ROTATIONAL TRANSFORM FORMULA

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} X_1 - X_0 \\ Y_1 - Y_0 \end{bmatrix} + \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

$X_0 \bullet Y_0$: Center Coordinate $X_1 \bullet Y_1$: Display Coordinate

$X_2 \bullet Y_2$: Coordinate before calculation of display coordinate

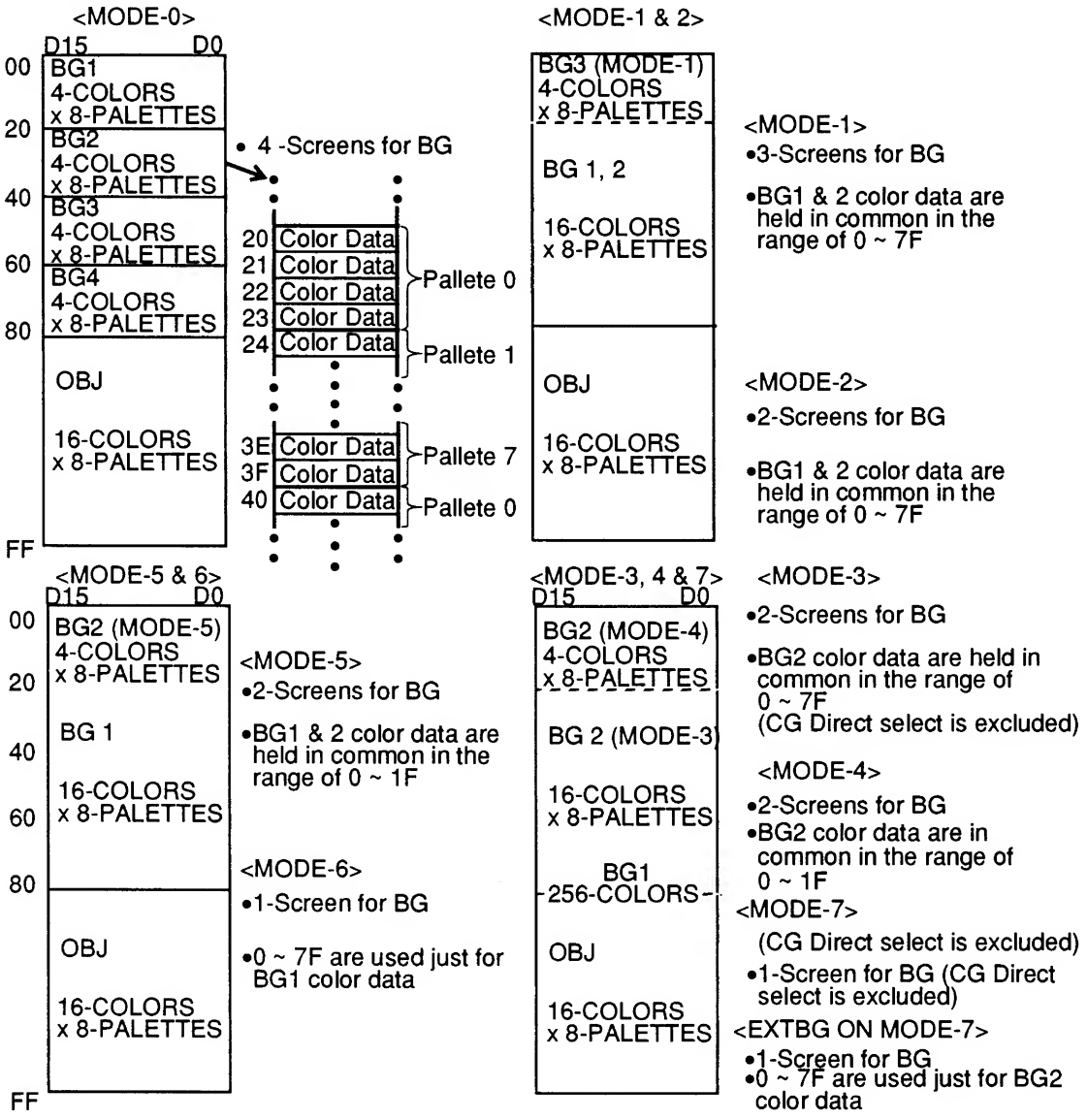
If the reduction rates for X-dir (a) and the reduction rates for Y-dir (β) are considered, the formula described above will be as follows:

$A=\cos\gamma \times (1/a)$, $B=\sin\gamma \times (1/a)$,

$C=-\sin\gamma \times (1/\beta)$, $D=\cos\gamma \times (1/\beta)$,

(NCL PG 80)

CG-RAM



CG-RAM COLOR DATA

BLUE					GREEN					RED				
d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0

DIRECT SELECT COLOR DATA

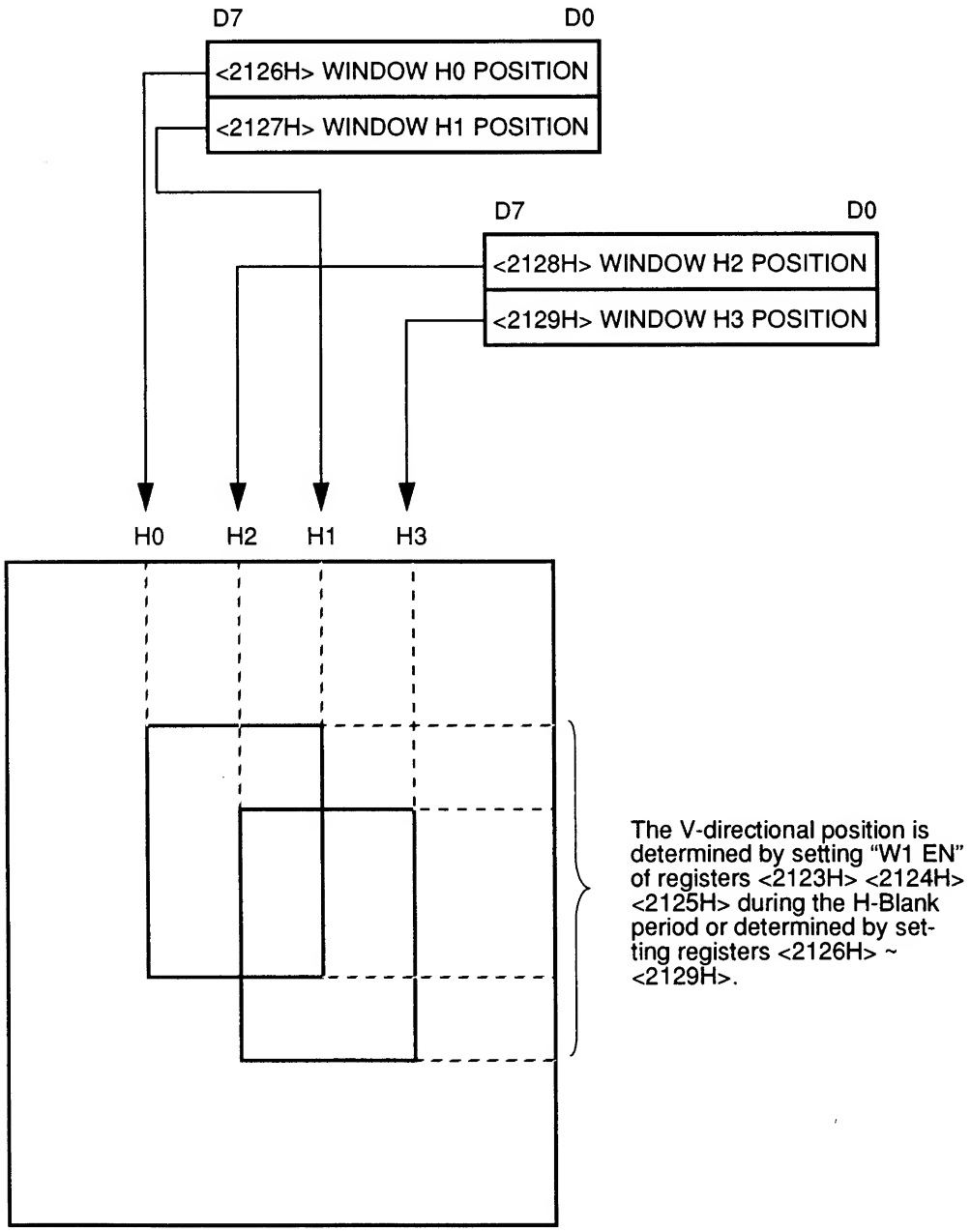
BLUE					GREEN					RED				
DA7	DA6	CL2	0	0	DA5	DA4	DA3	CL1	0	DA2	DA1	DA0	CL0	0

NOTE: DA0 ~ DA7 are used for the character dot data. CL0 ~ CL2 are used for the BG-SC data of the color. (However, in case of Mode-7, CL0 ~ CL2 should be "0")
NOTE: If they are "0," it becomes transparent. The color of CG-RAM address (00H) will be background.

(NCL PG 81)

WINDOW

REGISTER <2126H> ~ <2129H>

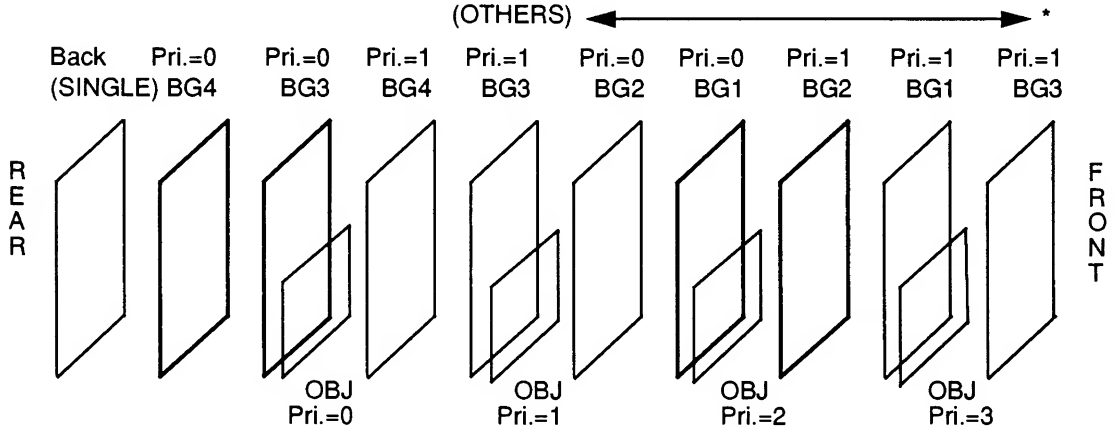


(NCL PG 82)

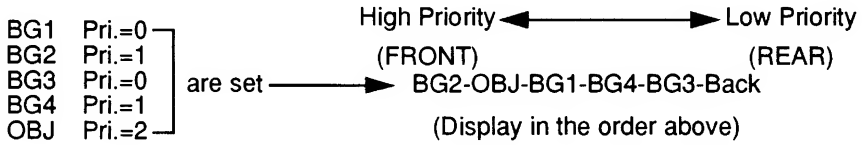
BG & OBJ PRIORITY

4-SCREEN/3-SCREEN MODE (In case Mode 0 and 1 are selected by register <2105H>)

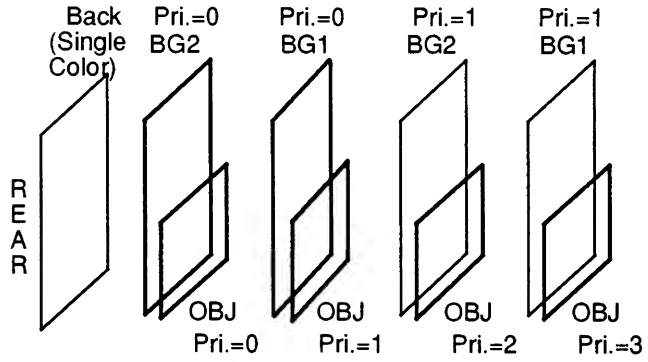
*In case "D3=1" is selected by register <2105H> in the mode-1



<Example of Display Priority (in case of mode 0)>



2-SCREEN/1-SCREEN MODE (in case Mode 2 ~ 7 is selected by register <2105H>)

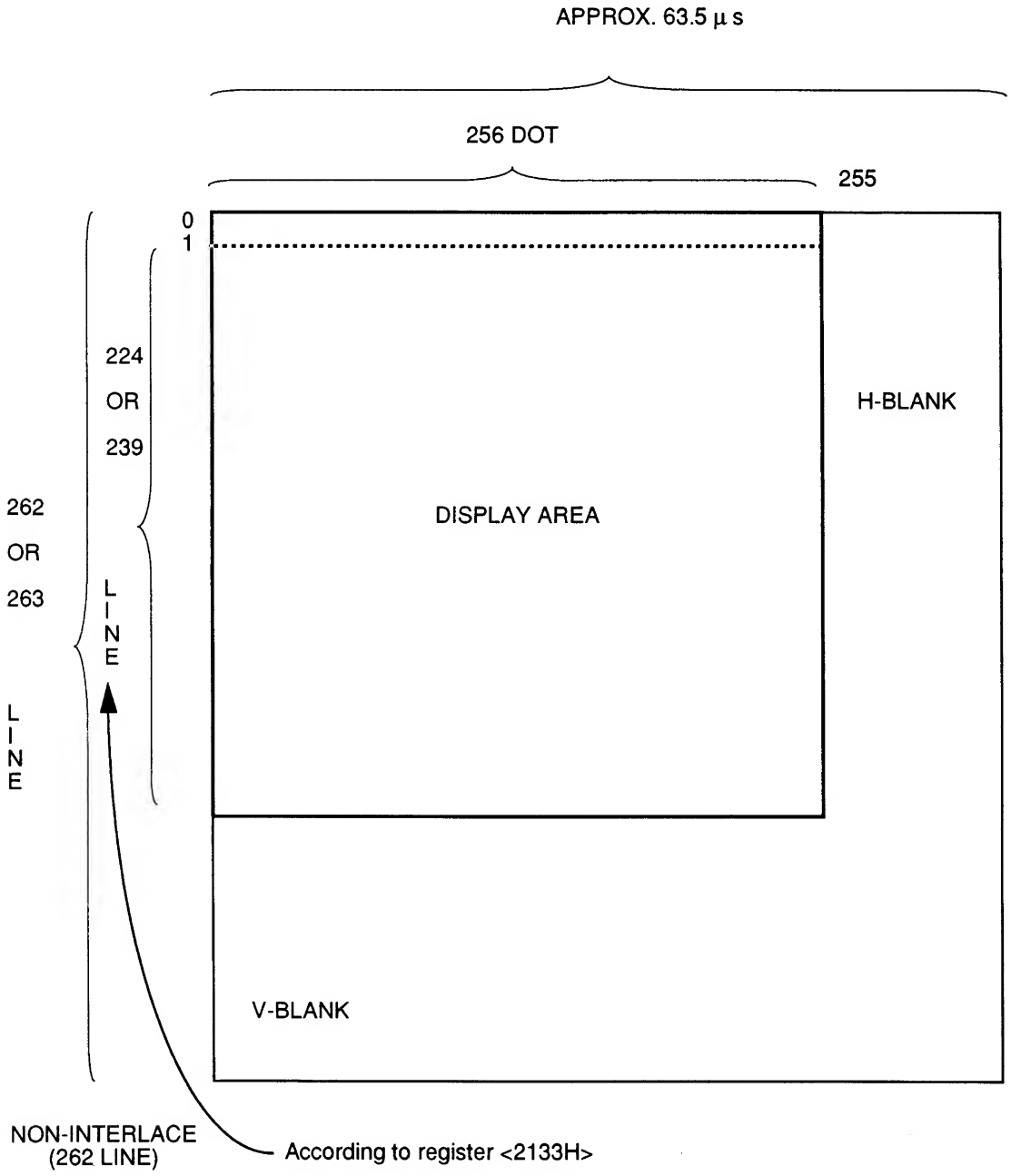


NOTE: In case of the display priority between the OBJ's, normally the lower numbered OBJ will be displayed as higher priority. (See page 1-20-2 for exception.)
This display priority will be determined before the priority between OBJ and BG is determined.

NOTE: In case of Mode 7, the priority is 0 at BG1.

(NCL PG 83)

SCREEN



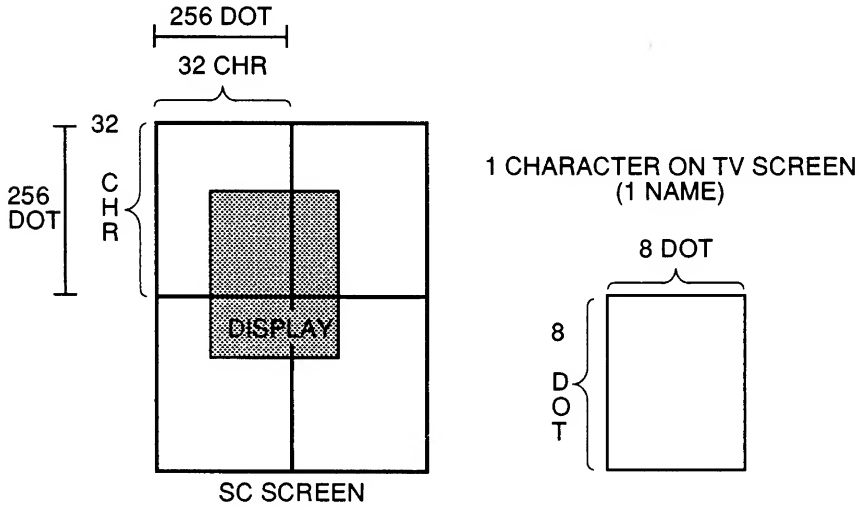
(NCL PG 84)

BG SCREEN

H/V SCROLL ① (Scroll range by the combination of modes and SC size against screen)
<Example: in case SC size is "3" - refer to register 2107H ~ 210AH>

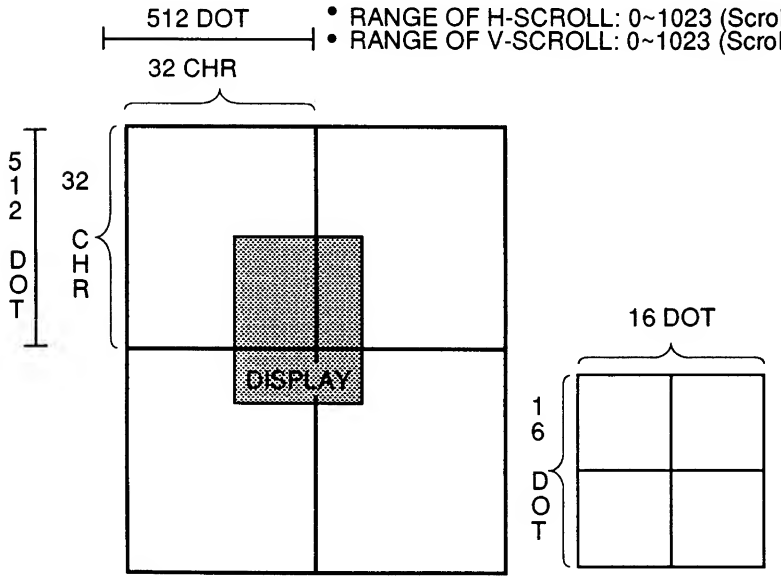
*In case of mode 0, 1, 2, 3, & 4

- BG SIZE (8 x 8)



- RANGE OF H-SCROLL: 0~511 (Scroll 1-dot to the left by adding 1)
- RANGE OF V-SCROLL: 0~511 (Scroll 1-dot up by adding 1)

- BG SIZE (16 x 16)



- RANGE OF H-SCROLL: 0~1023 (Scroll 1-dot to the left by adding 1)
- RANGE OF V-SCROLL: 0~1023 (Scroll 1-dot up by adding 1)

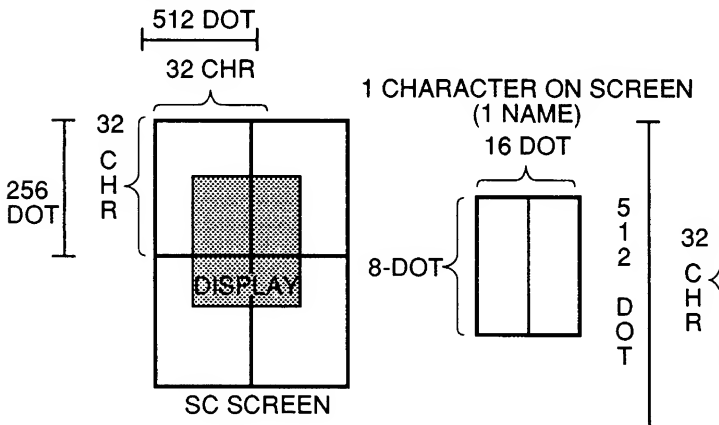
(NCL PG 85)

BG SCREEN

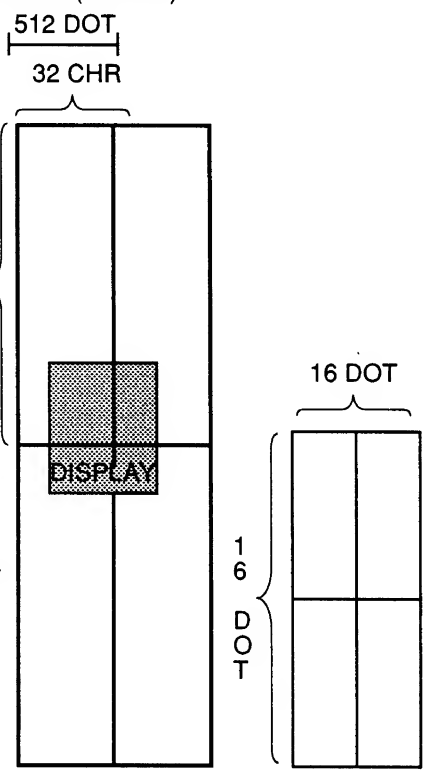
H/V SCROLL ② (Scroll range by the combination of modes and SC size against screen)
<Example: in case SC size is "3" - refer to register 2107H ~ 210AH>

*In case of MODE-5, MODE-6 & NON-INTERLACE

• BG SIZE (8 x 8)



• BG SIZE (16 X 16)

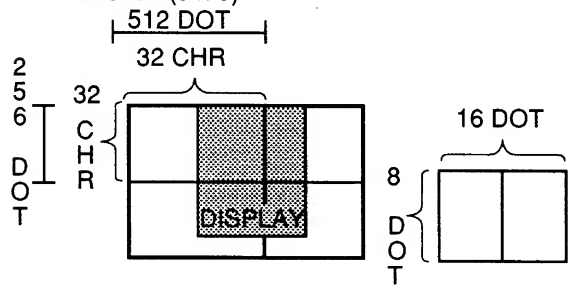


- RANGE OF H-SCROLL: 0~511 (Scroll 2-dots to the left by adding 1)
- RANGE OF V-SCROLL: 0~511 (Scroll 1-dot up by adding 1)

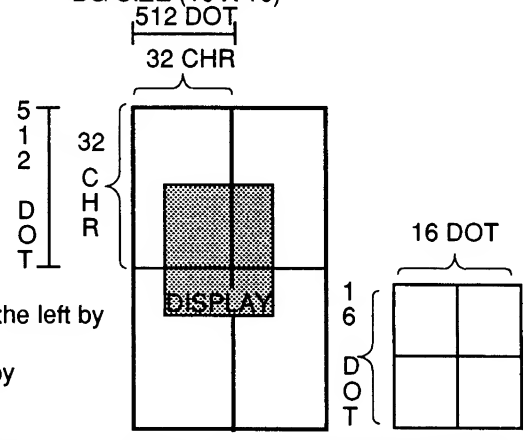
- RANGE OF H-SCROLL: 0~511 (Scroll 2-dots to the left by adding 1)
- RANGE OF V-SCROLL: 0~1023 (Scroll1-dot up by adding 1)

*In case of MODE-5, MODE-6 & INTERLACE

• BG SIZE (8 x 8)



• BG SIZE (16 X 16)

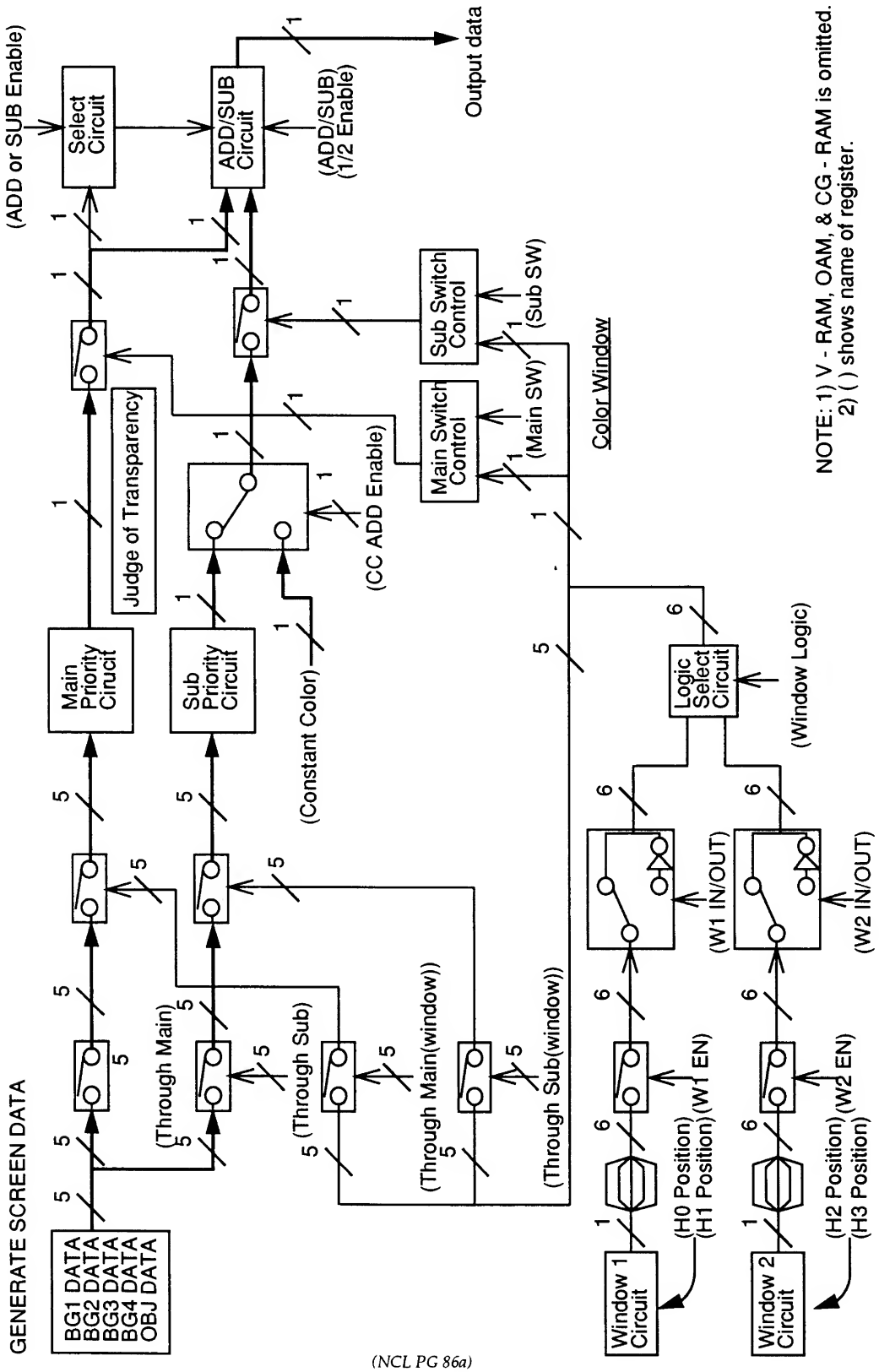


- RANGE OF H-SCROLL: 0~511 (Scroll 2-dots to the left by adding 1)
- RANGE OF V-SCROLL: 0~511 (Scroll 1-dot up by adding 1)

- RANGE OF H-SCROLL: 0~511 (Scroll 2-dots to the left by adding 1)
- RANGE OF V-SCROLL: 0~1023 (Scroll 1 dot up by adding 1)

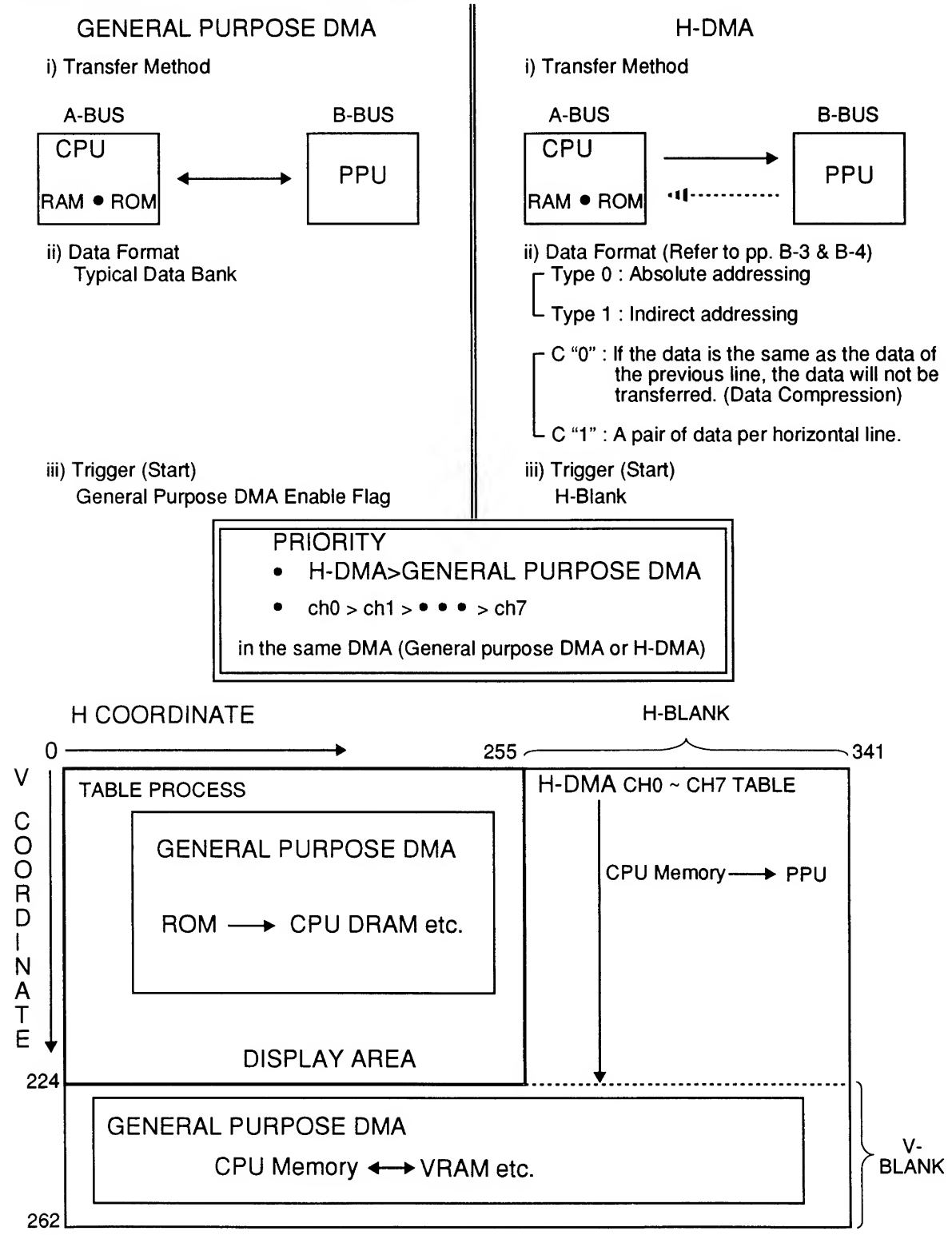
(NCL PG 86)

Figure A-1 SNES PPU Main/Sub Screen Window



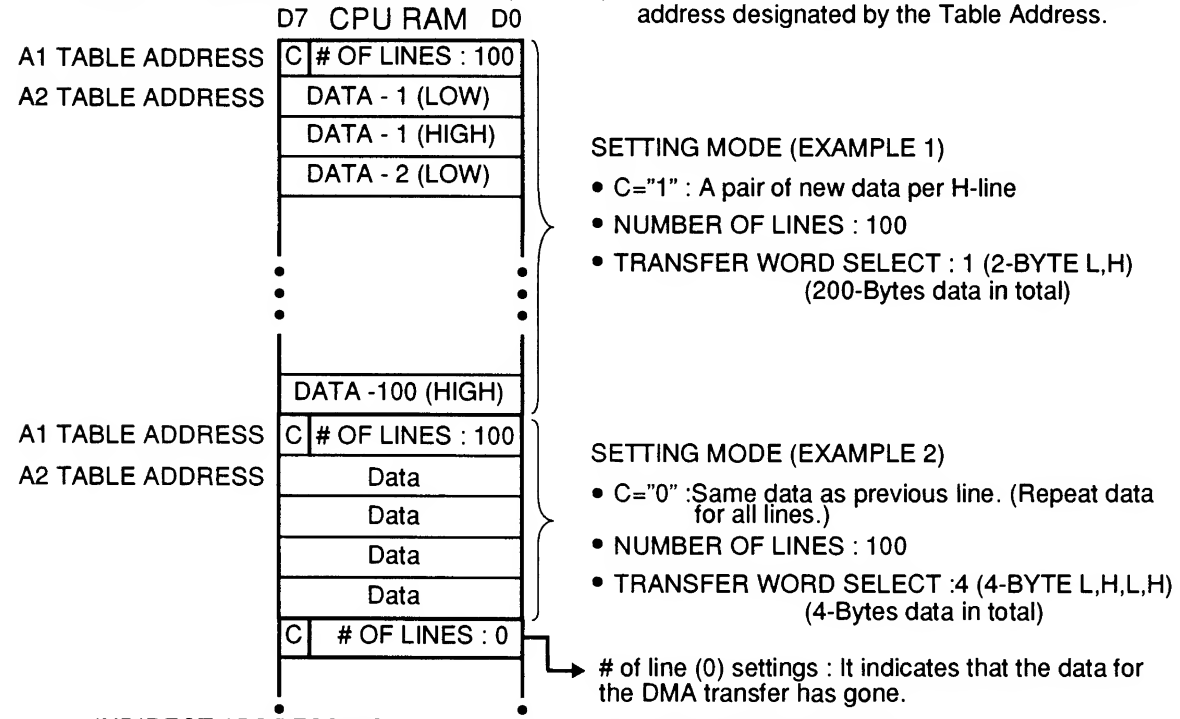
(NCL PG 86a)

Appendix B. CPU Registers

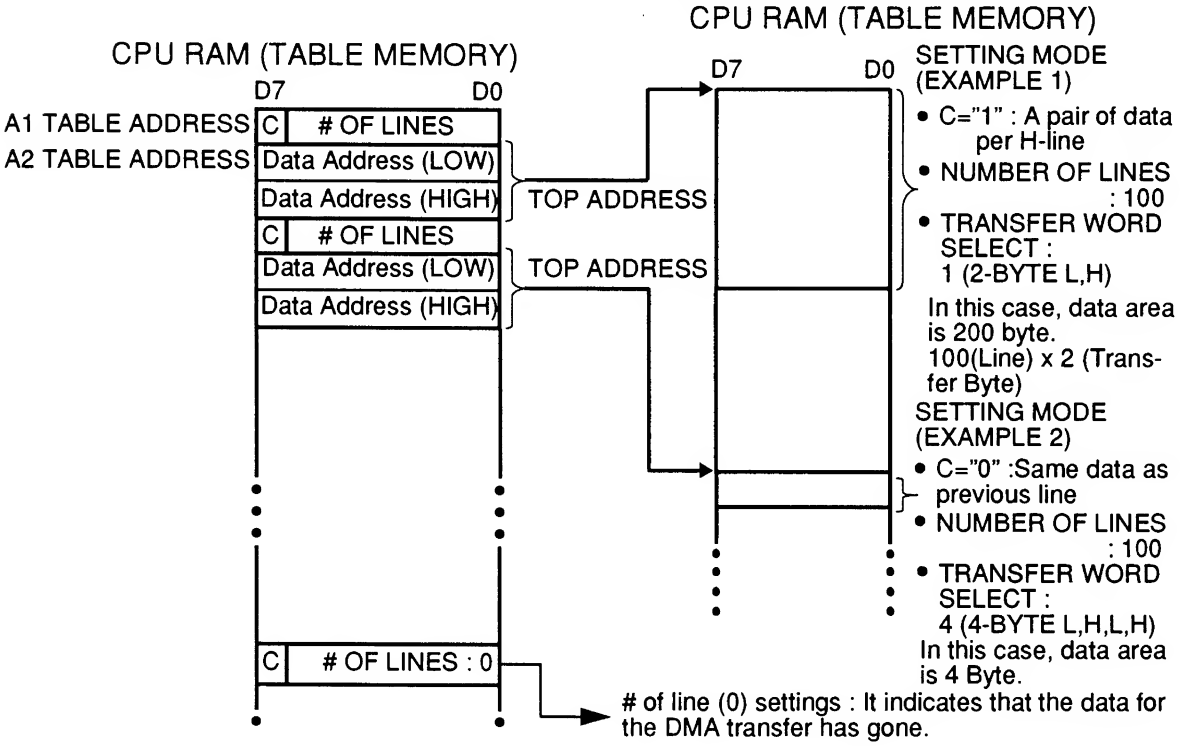


(NCL PG 101)

H-DMA ABSOLUTE ADDRESSING (TYPE-0) : This is a mode used to transfer the data of the address designated by the Table Address.

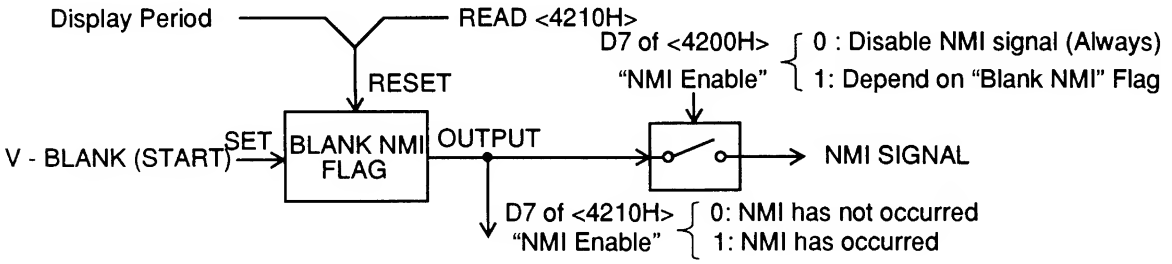


INDIRECT ADDRESSING (TYPE-1) : This is a mode used to transfer the data of the address designated by the Data Address, which is stored to the address designated by the Table Address.



(NCL PG 102)

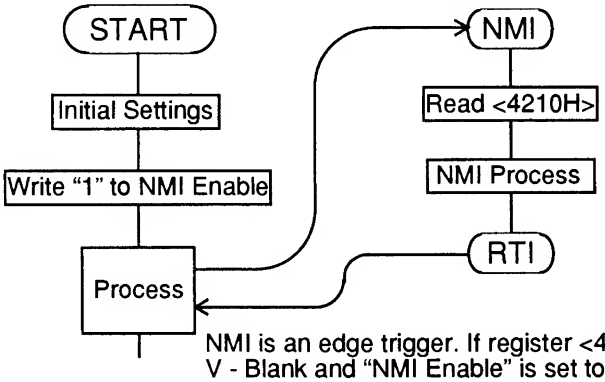
DETECT BEGINNING OF V - BLANK



The "Blank NMI" flag of register <4210H> will be set at the beginning of V - Blank and will reset at the end of V - Blank. It may also be reset by reading register <4210H>.

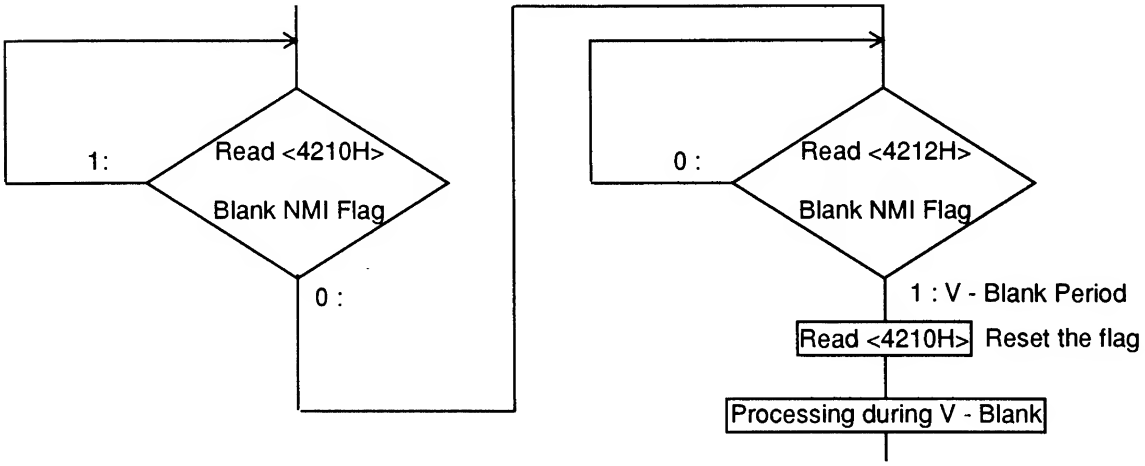
<EXAMPLE>

1. In case of detecting the beginning of V - Blank by NMI :



NMI is an edge trigger. If register <4210H> is not read during V - Blank and "NMI Enable" is set to "1", NMI will be duplicated.

2. In case of detecting the beginning of V - Blank by the flag :



(NCL PG 103)

SUMMARY OF REGISTERS

REGISTERS (WRITE) S - PPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2100H	Blanking				Fade IN/OUT (0 ~ 15)			
2101H	OBJ Size Select			OBJ Name Select	OBJ Name Base Address			
2102H	OAM Address							
2103H	OAM Priority Rotation							OAM Address MSB
2104H	OAM Data (Low, High)							
2105H	BG Size BG4	BG3	BG2	BG1	BG 3 Priority	BG Mode (0 ~ 7)		
2106H	Mosaic Size				Mosaic Enable BG4	BG3	BG2	BG1
2107H	BG1 SC Base Address						BG1 SC Size	
2108H	BG2 SC Base Address						BG2 SC Size	
2109H	BG3 SC Base Address						BG3 SC Size	
210AH	BG4 SC Base Address						BG4 SC Size	
210BH	BG2 Name Base Address				BG1 Name Base Address			
210CH	BG4 Name Base Address				BG3 Name Base Address			
210DH	BG1 H - Offset (Low, High)							
210EH	BG1V - Offset (Low, High)							
210FH	BG2 H - Offset (Low, High)							
2110H	BG2 V - Offset (Low, High)							
2111H	BG3 H - Offset (Low, High)							
2112H	BG3 V - Offset (Low, High)							
2113H	BG4 H - Offset (Low, High)							
2114H	BG4 V - Offset (Low, High)							
2115H	H/L Inc				V - RAM Address Sequence Mode Full Graphic		SC Increment	
2116H	V - RAM Address (Low)							
2117H	V - RAM Address (High)							
2118H	V - RAM Data (Low)							
2119H	V - RAM Data (High)							
211AH	Screen Over						Screen Flip V	H

(NCL PG 104)

REGISTERS (WRITE) S - PPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
211BH	Matrix Parameter A (Low, High)							
211CH	Matrix Parameter B (Low, High)							
211DH	Matrix Parameter C (Low, High)							
211EH	Matrix Parameter D (Low, High)							
211FH	Matrix Parameter X (Low, High)							
2120H	Matrix Parameter Y (Low, High)							
2121H	CG - RAM Address							
2122H	CG - RAM Data (Low, High)							
2123H	BG2 Window				BG1 Window			
	W2 EN	IN/OUT	W1 EN	IN/OUT	W2 EN	IN/OUT	W1 EN	IN/OUT
2124H	BG4 Window				BG3 Window			
	W2 EN	IN/OUT	W1 EN	IN/OUT	W2 EN	IN/OUT	W1 EN	IN/OUT
2125H	Color Window				OBJ Window			
	W2 EN	IN/OUT	W1 EN	IN/OUT	W2 EN	IN/OUT	W1 EN	IN/OUT
2126H	Window H0 Position (0 ~ 255)							
2127H	Window H1 Position (0 ~ 255)							
2128H	Window H2 Position (0 ~ 255)							
2129H	Window H3 Position (0 ~ 255)							
212AH	Window Logic							
	BG4		BG3		BG2		BG1	
212BH					Window Logic			
					Color		OBJ	
212CH				Through Main				
				OBJ	BG4	BG3	BG2	BG1
212DH				Through Sub				
				OBJ	BG4	BG3	BG2	BG1
212EH				Through Main (Window)				
				OBJ	BG4	BG3	BG2	BG1
212FH				Through Sub (Window)				
				OBJ	BG4	BG3	BG2	BG1
2130H	Window ON/OFF						CG ADD Enable	Direct Select
	Main SW (A)		Sub SW (B)					
2131H	ADD/SUB	1/2 Enable	ADD or SUB Enable					
			BACK	OBJ	BG4	BG3	BG2	BG1
2132H	Color Constant Data							
	Blue	Green	Red	Color Brilliance Data				
2133H	EXT. Sync.	EXT. Input			Pseudo 512	224/239	OBJ - V Select	Inter-lace

(NCL PG 104)

S - PPU READ REGISTER

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2134H	M P Y (Low)							
2135H	M P Y (Mid)							
2136H	M P Y (High)							
2137H	Soft Latch for H/V Counter							
2138H	OAM Data (Low, High)							
2139H	V - RAM Data (Low)							
213AH	V - RAM Data (High)							
213BH	CG Data (Low, High)							
213CH	Output Data of H - Counter (Low, High)							
213DH	Output Data of V - Counter (Low, High)							
213EH	Time Over	Range Over	Master /Slave		S - PPU1 Version Number			
213FH	Field	EXT. Latch		NTSC /PAL	S - PPU2 Version Number			

APU READ/WRITE REGISTER

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2140H	APU I/O Port							
2141H	APU I/O Port							
2142H	APU I/O Port							
2143H	APU I/O Port							

WORK RAM READ/WRITE REGISTER

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2180H	WORK RAM Data							

WORK RAM WRITE REGISTER

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
2181H	WORK RAM Address (Low)							
2182H	WORK RAM Address (Mid)							
2183H	WORK RAM Address (High)							

REGISTERS (WRITE) S - CPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
4200H	NMI Enable		Timer Enable V - EN	H- EN				Joy - C Enable
4201H	I/O Port							
4202H	Multiplicand - A							
4203H	Multiplier - B							
4204H	Dividend - C (Low)							
4205H	Dividend - C (High)							
4206H	Divisor - B							
4207H	H - Counter Timer							
4208H								H - MSB
4209H	V - Counter Timer							
420AH								V - MSB
420BH	General Purpose DMA (Enable Flag) CH7 EN	CH6 EN	CH5 EN	CH4 EN	CH3 EN	CH2 EN	CH1 EN	CH0 EN
420CH	H-DMA (Enable Flag) CH7 EN	CH6 EN	CH5 EN	CH4 EN	CH3 EN	CH2 EN	CH1 EN	CH0 EN
420DH								2.68 /3.58

(NCL PG 106)

REGISTERS (READ) S - CPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
4210H	Blank NMI				SNES - CPU Version Number			
4211H	Timer IRQ							
4212H	V -Blank	H -Blank						Joy - C Enable
4213H	I/O Port							
4214H	Quotient - A (Low)							
4215H	Quotient - A (High)							
4216H	Product - C / Remainder (Low)							
4217H	Product - C / Remainder (High)							
4218H	Joy Controller I (Low)							
4219H	Joy Controller I (High)							
421AH	Joy Controller II (Low)							
421BH	Joy Controller II (High)							
421CH	Joy Controller III (Low)							
421DH	Joy Controller III (High)							
421EH	Joy Controller IV (Low)							
421FH	Joy Controller IV (High)							

REGISTERS (WRITE) S - CPU

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
43X0H	CHX *T-Org	CHX Type		A - Bus Address INC/DEC	Fixed	CHX Transfer Word Select		
43X1H	CHX B - Address							
43X2H	CHX A1 Table Address (Low)							
43X3H	CHX A1 Table Address (High)							
43X4H	CHX A Table Bank							
43X5H	CHX Data Address (H-DMA) / Number of Bytes to be Transferred (General Purpose DMA) (Low)							
43X6H	CHX Data Address (H-DMA) / Number of Bytes to be Transferred (General Purpose DMA) (High)							
43X7H	CHX Data Bank (H - DMA)							
43X8H	CHX A2 Table Address (Low)							
43X9H	CHX A2 Table Address (High)							
43XAH	Continue	Number of Lines						

* T - Org means the "Transfer Orientation".

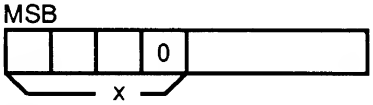
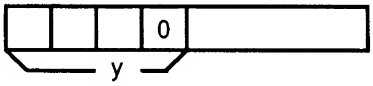
(NCL PG 106)

Appendix C SPC700 Commands

C.1 SUMMARY OF SPC700 COMMANDS

An SPC700 series is used for the SNES sound source CPU. However, standby and sleep modes cannot be used. The command set operand notation and explanation of command activity are indicated in the table below. The upper portion of the table contains symbols necessary to operand description. These are symbols necessary for assembler description. In the lower portion of the table, the values of the various operands are expressed as symbols. Assembler descriptions are given as numerical values or labels.

Table C-1 Command Operand Symbols and Meaning

Symbol	Meaning
A	A Register
X	X Register
Y	Y Register
PSW	Program Status Word
YA	Y, A paired 16-bit register
PC	Program Counter
SP	Stack Pointer
( )	Indirect Expression
( )+	Indirect Auto-increment Expression
#	Immediate Data
!	Absolute Address
/	Bit Reversal
.	Bit Position Indicator
[ ]	Indexed Indirect Expression
H	Hexadecimal Notation
imm	8-bit Immediate Data
dp	Offset Address within Direct Page
abs	16-bit Absolute Address
rel	Relative Offset 2's Complement
mem	Boolean Bit Operation Address
bit	Bit Location
x	MSB 
y	MSB 
upage	Offset Within U Page
n	Vector Call Number

(NCL PG 35)

The following symbols are used, in addition to those on the previous page, for the purpose of explaining operational functions.

Table C-2 Symbols and Meaning for Operational Description

Symbol	Meaning
N	Negative Flag
V	Overflow Flag
P	Direct Page Flag
B	Break Flag
H	Half Carry Flag
I	Indirect Master Enable Flag
Z	Zero Flag
C	Carry Flag
+	Addition
-	Subtraction
:	Comparison
AND	Logic Product
OR	Logic Sum
EOR	Exclusive Logic Sum
*	Multiplication
/	Division
Q	Division Quotient
R	Division Remainder
<d>	Destination
<S>	Source
→	Direction of Data Transmission
- -	Data Decrement
+ +	Data Increment
< <	1 Bit Shift Left
> >	1 Bit Shift Right

Note: The number of cycles of conditional branching commands are appropriate to cases when there is no branching to the left side and there is branching to the right side.

Table C-3 Explaination of Symbols in the Status Flag Column

Symbol	Meaning
.	No Change
0	Cleared to "0"
1	Set to "1"
Flag Name	Set or Cleared Depending on Result

(NCL PG 36)

Table C-4 8-bit Data Transmission Commands, Group 1

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
MOV	A, #imm	E8	2	2	A ← imm	N.....Z
MOV	A. (X)	E6	1	3	A ← (X)	N.....Z
MOV	A, (X)+	BF	1	4	A ← (X) with auto increment	N.....Z
MOV	A, dp	E4	2	3	A ← (dp)	N.....Z
MOV	A, dp+X	F4	2	4	A ← (dp+X)	N.....Z
MOV	A, !abs	E5	3	4	A ← (abs)	N.....Z
MOV	A, !abs+X	F5	3	5	A ← (abs+X)	N.....Z
MOV	A, !abs+Y	F6	3	5	A ← (abs+Y)	N.....Z
MOV	A, [dp+X]	E7	2	6	A ← ((dp+X+1)(dp+X))	N.....Z
MOV	A, [dp]+Y	F7	2	6	A ← ((dp+1)(dp)+Y)	N.....Z
MOV	X, #imm	CD	2	2	X ← imm	N.....Z
MOV	X, dp	F8	2	3	X ← (dp)	N.....Z
MOV	X, dp+Y	F9	2	4	X ← (dp+Y)	N.....Z
MOV	X, !abs	E9	3	4	X ← (abs)	N.....Z
MOV	Y, #imm	8D	2	2	Y ← imm	N.....Z
MOV	Y, dp	EB	2	3	Y ← (dp)	N.....Z
MOV	Y, dp+X	FB	2	4	Y ← (dp+X)	N.....Z
MOV	Y, !abs	EC	3	4	Y ← (abs)	N.....Z

Table C-5 8-bit Data Transmission Commands, Group 2

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
MOV	(X), A	C6	1	4	A → (X)	.....
MOV	(X)+, A	AF	1	4	A → (X) with auto increment	.....
MOV	dp, A	C4	2	4	A → (dp)	.....
MOV	dp+X, A	D4	2	5	A → (dp+X)	.....
MOV	!abs, A	C5	3	5	A → (abs)	.....
MOV	!abs+X, A	D5	3	6	A → (abs+X)	.....
MOV	!abs+Y, A	D6	3	6	A → (abs+Y)	.....
MOV	[dp+X], A	C7	2	7	A → ((dp+X+1)(dp+X))	.....
MOV	[dp]+Y, A	D7	2	7	A → ((dp+1)(dp)+Y)	.....
MOV	dp, X	D8	2	4	X → (dp)	.....
MOV	dp+Y, X	D9	2	5	X → (dp+Y)	.....
MOV	!abs, X	C9	3	5	X → (abs)	.....
MOV	dp, Y	CB	2	4	Y → (dp)	.....
MOV	dp+X, Y	DB	2	5	Y → (dp+X)	.....
MOV	!abs, Y	CC	3	5	Y → (abs)	.....

(NCL PG 37)

Table C-6 8-bit Data Transmission Commands, Group 3

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
MOV	A, X	7D	1	2	A← X	N.....Z
MOV	A, Y	DD	1	2	A← Y	N.....Z
MOV	X, A	5D	1	2	X← A	N.....Z
MOV	Y, A	FD	1	2	Y← A	N.....Z
MOV	X, SP	9D	1	2	X← SP	N.....Z
MOV	SP, X	BD	1	2	SP← X	.....
MOV	dp<d>, dp<s>	FA	3	5	(dp<d>) ← (dp<s>)	.....
MOV	dp, #imm	8F	3	5	(dp)← imm	.....

Table C-7 8-bit Arithmetic Operation Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
ADC	A, #imm	88	2	2	A ← A + imm + C	NV..H.ZC
ADC	A, (X)	86	1	3	A ← A + (X) + C	NV..H.ZC
ADC	A, dp	84	2	3	A ← A + (dp) + C	NV..H.ZC
ADC	A, dp+X	94	2	4	A ← A + (dp+X) + C	NV..H.ZC
ADC	A, !abs	85	3	4	A ← A + (abs) + C	NV..H.ZC
ADC	A, !abs+X	95	3	5	A ← A + (abs + X) + C	NV..H.ZC
ADC	A, !abs+Y	96	3	5	A ← A + (abs + Y) + C	NV..H.ZC
ADC	A, [dp+X]	87	2	6	A ← A + (dp+X+1)(dp+X) + C	NV..H.ZC
ADC	A, [dp]+Y	97	2	6	A ← A + ((dp+1)(dp)+Y) + C	NV..H.ZC
ADC	(X), (Y)	99	1	5	(X)← (X) + (Y) + C	NV..H.ZC
ADC	dp<d>, dp<s>	89	3	6	(dp<d>)← (dp<d>) + (dp<s>) + C	NV..H.ZC
ADC	dp, #imm	98	3	5	(dp)← (dp) + imm + C	NV..H.ZC
SBC	A, #imm	A8	2	2	A ← A - imm- $\overline{C}$	NV..H.ZC
SBC	A, (X)	A6	1	3	A ← A - (X)- $\overline{C}$	NV..H.ZC
SBC	A, dp	A4	2	3	A ← A - (dp)- $\overline{C}$	NV..H.ZC
SBC	A, dp+X	B4	2	4	A ← A - (dp+X)- $\overline{C}$	NV..H.ZC
SBC	A, !abs	A5	3	4	A ← A - (abs)- $\overline{C}$	NV..H.ZC
SBC	A, !abs+X	B5	3	5	A ← A - (abs + X)- $\overline{C}$	NV..H.ZC
SBC	A, !abs+Y	B6	3	5	A ← A - (abs + Y)- $\overline{C}$	NV..H.ZC
SBC	A, [dp+X]	A7	2	6	A ← A - (dp+X+1)(dp+X) - $\overline{C}$	NV..H.ZC
SBC	A, [dp]+Y	B7	2	6	A ← A - ((dp+1)(dp)+Y) - $\overline{C}$	NV..H.ZC
SBC	(X), (Y)	B9	1	5	(X)← (X)- (Y)- $\overline{C}$	NV..H.ZC
SBC	dp<d>, dp<s>	A9	3	6	(dp<d>)← (dp<d>) - (dp<s>)- $\overline{C}$	NV..H.ZC
SBC	dp, #imm	B8	3	5	(dp)← (dp) - imm - $\overline{C}$	NV..H.ZC
CMP	A, #imm	68	2	2	A - imm	N.....ZC
CMP	A, (X)	66	1	3	A - (X)	N.....ZC
CMP	A, dp	64	2	3	A -(dp)	N.....ZC
CMP	A, dp+X	74	2	4	A - (dp+X)	N.....ZC
CMP	A, !abs	65	3	4	A - (abs)	N.....ZC
CMP	A, !abs+X	75	3	5	A - (abs+X)	N.....ZC
CMP	A, !abs+Y	76	3	5	A - (abs+Y)	N.....ZC
CMP	A, [dp+X]	67	2	6	A - ((dp+X+1)(dp+X))	N.....ZC
CMP	A, [dp]+Y	77	2	6	A - ((dp+1)(dp)+Y)	N.....ZC
CMP	(X), (Y)	79	1	5	(X) - (Y)	N.....ZC
CMP	dp<d>, dp<s>	69	3	6	(dp<d>) - (dp<s>)	N.....ZC
CMP	dp, #imm	78	3	5	(dp) - imm	N.....ZC
CMP	X, #imm	C8	2	2	X - imm	N.....ZC
CMP	X, dp	3E	2	3	X - (dp)	N.....ZC
CMP	X, !abs	1E	3	4	X - (abs)	N.....ZC
CMP	Y, #imm	AD	2	2	Y -imm	N.....ZC
CMP	Y, dp	7E	2	3	Y -(dp)	N.....ZC
CMP	Y, !abs	5E	3	4	Y -(abs)	N.....ZC

(NCL PG 39)

Table C-8 8-bit Logic Operation Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
AND	A, #imm	28	2	2	A ← A AND imm	N.....Z.
AND	A, (X)	26	1	3	A ← A AND (X)	N.....Z.
AND	A, dp	24	2	3	A ← A AND (dp)	N.....Z.
AND	A, dp+X	34	2	4	A ← A AND (dp+X)	N.....Z.
AND	A, !abs	25	3	4	A ← A AND (abs)	N.....Z.
AND	A, !abs+X	35	3	5	A ← A AND (abs+X)	N.....Z.
AND	A, !abs+Y	36	3	5	A ← A AND (abs+Y)	N.....Z.
AND	A, [dp+X]	27	2	6	A ← A AND ((dp+X+1)(dp+X))	N.....Z.
AND	A, [dp]+Y	37	2	6	A ← A AND ((dp+1)(dp)+Y)	N.....Z.
AND	(X), (Y)	39	1	5	(X) ← (X) AND (Y)	N.....Z.
AND	dp<d>, dp<s>	29	3	6	(dp<d>) ← (dp<d>) AND (dp<s>)	N.....Z.
AND	dp, #imm	38	3	5	(dp) ← (dp) AND imm	N.....Z.
OR	A, #imm	08	2	2	A ← A OR imm	N.....Z.
OR	A, (X)	06	1	3	A ← A OR (X)	N.....Z.
OR	A, dp	04	2	3	A ← A OR (dp)	N.....Z.
OR	A, dp+X	14	2	4	A ← A OR (dp+X)	N.....Z.
OR	A, !abs	05	3	4	A ← A OR (abs)	N.....Z.
OR	A, !abs+X	15	3	5	A ← A OR (abs+X)	N.....Z.
OR	A, !abs+Y	16	3	5	A ← A OR (abs+Y)	N.....Z.
OR	A, [dp+X]	07	2	6	A ← A OR ((dp+X+1)(dp+X))	N.....Z.
OR	A, [dp]+Y	17	2	6	A ← A OR ((dp+1)(dp)+Y)	N.....Z.
OR	(X), (Y)	19	1	5	(X) ← (X) OR (Y)	N.....Z.
OR	dp<d>, dp<s>	09	3	6	(dp<d>) ← (dp<d>) OR (dp<s>)	N.....Z.
OR	dp, #imm	18	3	5	(dp) ← (dp) OR imm	N.....Z.
EOR	A, #imm	48	2	2	A ← A EOR imm	N.....Z.
EOR	A, (X)	46	1	3	A ← A EOR (X)	N.....Z.
EOR	A, dp	44	2	3	A ← A EOR (dp)	N.....Z.
EOR	A, dp+X	54	2	4	A ← A EOR (dp+X)	N.....Z.
EOR	A, !abs	45	3	4	A ← A EOR (abs)	N.....Z.
EOR	A, !abs+X	55	3	5	A ← A EOR (abs+X)	N.....Z.
EOR	A, !abs+Y	56	3	5	A ← A EOR (abs+Y)	N.....Z.
EOR	A, [dp+X]	47	2	6	A ← A EOR ((dp+X+1)(dp+X))	N.....Z.
EOR	A, [dp]+Y	57	2	6	A ← A EOR ((dp+1)(dp)+Y)	N.....Z.
EOR	(X), (Y)	59	1	5	(X) ← (X) EOR (Y)	N.....Z.
EOR	dp<d>, dp<s>	49	3	6	(dp<d>) ← (dp<d>) EOR(dp<s>)	N.....Z.
EOR	dp, #imm	58	3	5	(dp) ← (dp) EOR imm	N.....Z.

(NCL PG 40)

Table C-9 Addition and Subtraction Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
INC	A	BC	1	2	++ A	N.....Z.
INC	dp	AB	2	4	++ (dp)	N.....Z.
INC	dp+X	BB	2	5	++ (dp+X)	N.....Z.
INC	!abs	AC	3	5	++ (abs)	N.....Z.
INC	X	3D	1	2	++ X	N.....Z.
INC	Y	FC	1	2	++ Y	N.....Z.
DEC	A	9C	1	2	-- A	N.....Z.
DEC	dp	8B	2	4	-- (dp)	N.....Z.
DEC	dp+X	9B	2	5	-- (dp+X)	N.....Z.
DEC	!abs	8C	3	5	-- (abs)	N.....Z.
DEC	X	1D	1	2	-- X	N.....Z.
DEC	Y	DC	1	2	-- Y	N.....Z.

Table C-10 Shift Rotation Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation		NVPBHIZC
ASL	A	1C	1	2	C << A	<<0	N.....ZC
ASL	dp	0B	2	4	C << (dp)	<<0	N.....ZC
ASL	dp+X	1B	2	5	C << (dp+X)	<<0	N.....ZC
ASL	!abs	0C	3	5	C << (abs)	<<0	N.....ZC
LSR	A	5C	1	2	C << A	<<C	N.....ZC
LSR	dp	4B	2	4	C << (dp)	<<C	N.....ZC
LSR	dp+X	5B	2	5	C << (dp+X)	<<C	N.....ZC
LSR	!abs	4C	3	5	C << (abs)	<<C	N.....ZC
ROL	A	3C	1	2	C << A	<<C	N.....ZC
ROL	dp	2B	2	4	C << (dp)	<<C	N.....ZC
ROL	dp+X	3B	2	5	C << (dp+X)	<<C	N.....ZC
ROL	!abs	2C	3	5	C << (abs)	<<C	N.....ZC
ROR	A	7C	1	2	C << A	<<C	N.....ZC
ROR	dp	6B	2	4	C << (dp)	<<C	N.....ZC
ROR	dp+X	7B	2	5	C << (dp+X)	<<C	N.....ZC
ROR	!abs	6C	3	5	C << (abs)	<<C	N.....ZC
XCN	A	9F	1	5	A (7 ~ 4) <-> A (3 ~ 0)		N.....Z.

Table C-11 16-bit Data Transmission Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
MOVW	YA,dp	BA	2	5	YA ← (dp+1)(dp)	N.....Z.
MOVW	dp, YA	DA	2	4	(dp+1)(dp) ← YA	.....

Table C-12 16-bit Operation Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
INCW	dp	3A	2	6	increment dp memory pair	N.....Z.
DECW	dp	1A	2	6	decrement dp memory pair	N.....Z.
ADDW	YA, dp	7A	2	5	YA ← YA+ (dp+1)(dp)	NV..H.ZC
SUBW	YA, dp	9A	2	5	YA ← YA- (dp+1)(dp)	NV..H.ZC
CMPW	YA, dp	5A	2	4	YA- (dp+1)(dp)	N.....ZC

Table C-13 Multiplication and Division Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
MUL	YA	CF	1	9	YA(16bits) ← Y * A	N.....Z.
DIV	YA, X	9E	1	12	Q:A R:Y ← YA/ X	NV..H.Z.

Table C-14 Decimal Compensation Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
DAA	A	DF	1	3	decimal adjust for addition	N.....ZC
DAS	A	BE	1	3	decimal adjust for subtraction	N.....ZC

Table C-15 Branching Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
BRA	rel	2F	2	4	branch always	.....
BEQ	rel	F0	2	2/4	branch on Z=1	.....
BNE	rel	D0	2	2/4	branch on Z=0	.....
BCS	rel	B0	2	2/4	branch on C=1	.....
BCC	rel	90	2	2/4	branch on C=0	.....
BVS	rel	70	2	2/4	branch on V=1	.....
BVC	rel	50	2	2/4	branch on V=0	.....
BMI	rel	30	2	2/4	branch on N=1	.....
BPL	rel	10	2	2/4	branch on N=0	.....
BBS	dp,bit, rel	x3	3	5/7	branch on dp, bit=1	.....
BBC	dp,bit, rel	y3	3	5/7	branch on dp, bit=0	.....
CBNE	dp,rel	2E	3	5/7	compare A with (dp) then BNE	.....
CBNE	dp+X, rel	DE	3	6/8	compare A with (dp+X) then BNE	.....
DBNZ	dp,rel	6E	3	5/7	decrement memory (dp) then JNZ	.....
DBNZ	Y,rel	FE	2	4/6	decrement Y then JNZ	.....
JMP	!abs	5F	3	3	jump to new location	.....
JMP	[!abs+X]	1F	3	6	PC ← (abs+X+1)(abs+X)	.....

(NCL PG 42)

Table C-16 Subroutine Call, Return Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
CALL	!abs	3F	3	8	subroutine call	.....
PCALL	upage	4F	2	6	upage call	.....
TCALL	n	n1	1	8	table call	.....
BRK		0F	1	8	software interrupt	...1.0..
RET		6F	1	5	return from subroutine	.....
RETI		7F	1	6	return from interrupt	(Restored)

Table C-17 Stack Operation Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
PUSH	A	2D	1	4	push A to stack	.....
PUSH	X	4D	1	4	push X to stack	.....
PUSH	Y	6D	1	4	push Y to stack	.....
PUSH	PSW	0D	1	4	push PSW to stack	.....
POP	A	AE	1	4	pop A from stack	.....
POP	X	CE	1	4	pop X from stack	.....
POP	Y	EE	1	4	pop Y from stack	.....
POP	PSW	8E	1	4	pop PSW from stack	(Restored)

Table C-18 Bit Operation Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
SET1	dip. bit	x2	2	4	set direct page bit	.....
CLR1	dip. bit	y2	2	4	clear direct page bit	.....
TSET1	!abs	0E	3	6	test and set bits with A	N.....Z.
TCLR1	!abs	4E	3	6	test and clear bits with A	N.....Z.
AND1	C, mem. bit	4A	3	4	C ← C AND (mem. bit)	C
AND1	C, /mem. bit	6A	3	4	C ← C AND (mem. bit)	C
OR1	C, mem. bit	0A	3	5	C ← C OR (mem. bit)	C
OR1	C, /mem. bit	2A	3	5	C ← C OR (mem. bit)	C
EOR1	C, mem. bit	8A	3	5	C ← C EOR (mem. bit)	C
NOT1	mem. bit	EA	3	5	complement (mem. bit)	.....
MOV1	C, mem. bit	AA	3	4	C ← (mem. bit)	C
MOV1	mem. bit, C	CA	3	6	C → (mem. bit)	.....

(NCL PG 43)

Table C-19 Program Status Flag Operation Commands

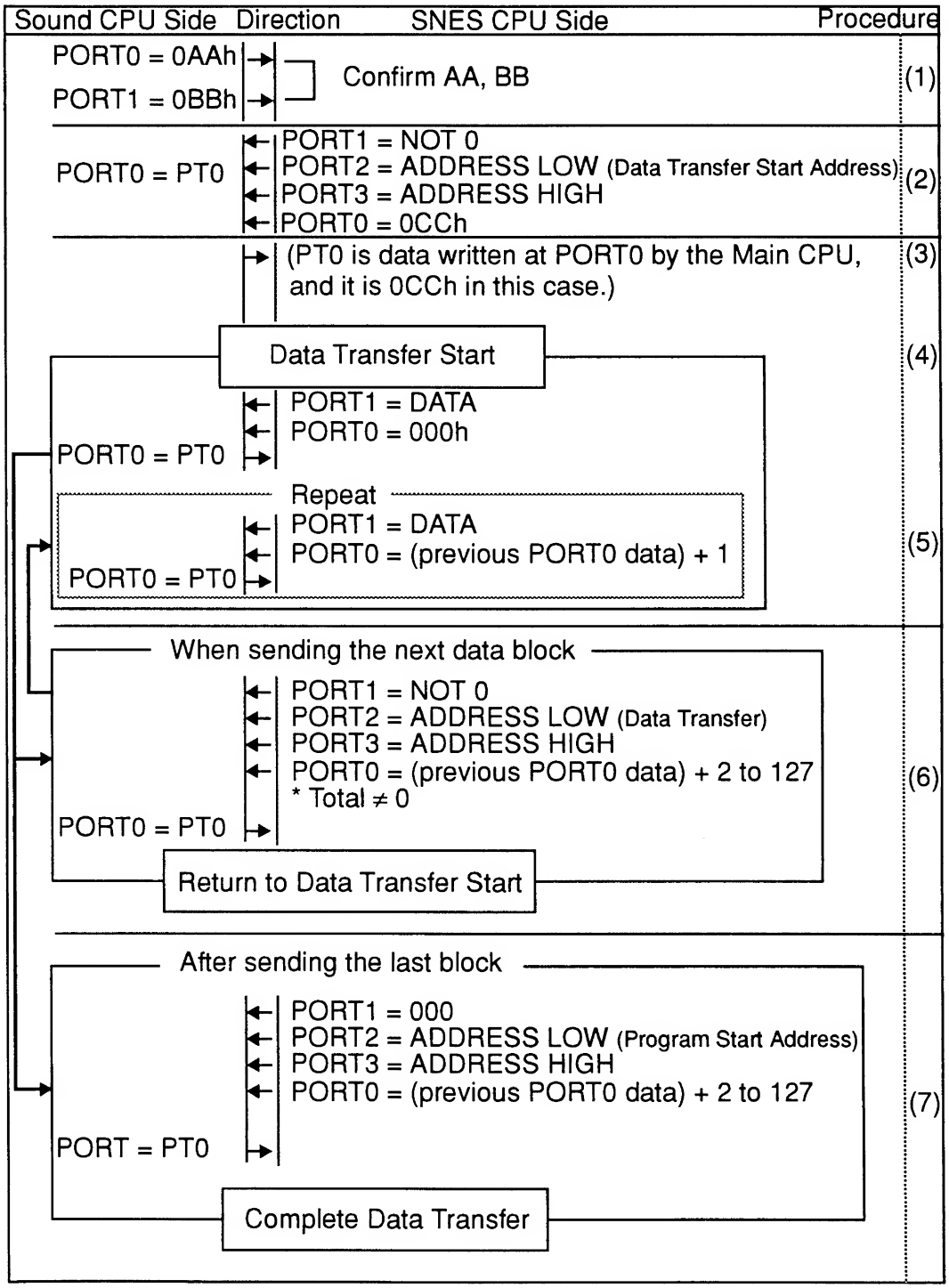
Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
CLRC		60	1	2	clear carry flag	0
SETC		80	1	2	set carry flag	1
NOTC		ED	1	3	complement carry flag	C
CLRV		E0	1	2	clear V and II	.0..0...
CLRP		20	1	2	clear direct page flag	..0.....
SETP		40	1	2	set direct page flag	..1.....
EI		A0	1	3	set interrupt enable flag	.....1.
DI		C0	1	3	clear interrupt enable flag	.....0.

Table C-20 Other Commands

Mnemonic	Operand	Code	Bytes	Cycles	Operation	NVPBHIZC
NOP		00	1	2	no operation	.....
SLEEP		EF	1	3	standby SLEEP mode	.....
STOP		FF	1	3	standby STOP mode	.....

Appendix D. Data Transfer Procedure

D.1 Data Transfer Procedure



(NCL PG 45)

D.2 Data Transfer Instruction

The transfer program on the Sound CPU is stored in the internal ROM called IPL ROM. This ROM functions after reset. The program ROM functions using the Main CPU and PORT 0 through 3.

(5) The sound CPU writes AAh to PORT 1. The Main CPU reads and confirms data at PORT 0 and 1.

(6) The Main CPU writes Start Address to PORT 2 and 3. After storing Port 2 and 3, store any number except 0 to PORT 1 and store CCh to PORT 0.

(7) The sound CPU checks PORT 0 for CCh and writes CCh to PORT 0.

(8) Start data transfer. The Main CPU writes first data to PORT 1 and writes 00h to PORT 0. The Sound CPU reads data from PORT 1 and writes 00h to PORT 0.

(9) The Main CPU checks PORT 0, writes next data to PORT 1, and increments of PORT 0. This is the data transfer procedure. The data block contains the quantity of data to be transferred.

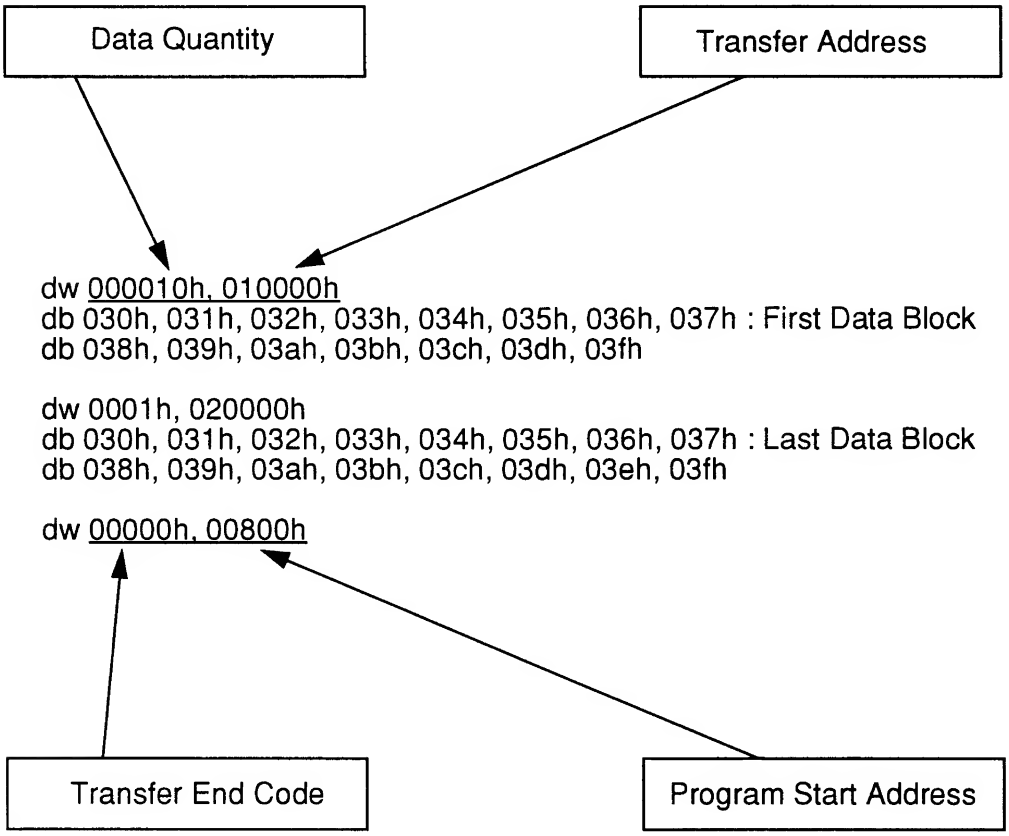
(10) When PORT 0 stops incrementing, proceed to the next step. The value that the SNES CPU writes to PORT 0 must not be 00h. Write any value but 00h to PORT 1. The Sound CPU writes the same value to PORT 0 and then returns to step (4).

(11) After sending all data blocks using steps (4) through (6), the data transfer is completed. Program Start Address is stored to PORT 2 and 3, Write 00h to PORT 1.

D.3 Data Block Organization

Data is divided into several blocks having consecutive addresses. The quantity of data (2 byte) and address (2 byte) are stored in front of data.

Data Block Example:



D.4 Sound Boot Loader V1.1

```
                    glb     Boot_APU
APU_port    0       equ     02140h
APU_port    1       equ     02141h
APU_port    2       equ     02142h
APU_port    3       equ     02143h
address             equ     00000h          Input Sound ROM Start Address
;                                           (3 byte) in 0 page and call
;           code                            "Boot_APU" from main routine.
Boot_APU    php
            rep     #00110000b
            idx16                   ;sony news
            mem16                   ;sony news
            on16i                   ;SNES Emulator
            on16a                   ;SNES Emulator
            ldy     #0
            lda     #0bbaah
boot_initial cmp    ! APU_PORT0     ;m16
            bne     boot_initial
            sep     #00100000b
            mem8                    ;sony news
            off16a                  ;SNES Emulator
            lda     #0cch
            bra     boot_entry1
boot_repeat lda     [address],y
            iny
            xba
            lda     #0
            bra     boot_entry2
boot_loop   xba
            lda     [address],y
            iny
            xba
boot_wait1  cmp     ! APU_PORT0
            bne     boot_wait1
            inc     a
boot_entry2 rep     #00100000b
            sta     ! APU_PORT0     ;m16
            sep     #00100000b
            dex
            bne     boot_loop
boot_wait2  cmp     ! APU_PORT0
            bne     boot_wait2
boot_zero   adc     #3
            beq     boot_zero
boot_entry1 pha                     ;>
            rep     #00100000b
            lda     [address],y     ;m16
            iny
            iny
            tax
            lda     [address],y     ;m16
            iny
            iny
            sta     ! APU_PORT2     ;m16
            sep     #00100000b
            cpx     #1
            lda     #0
            rol     a
            sta     !APU_PORT1
            adc     #07fh
            pla                     ;>
            sta     ! APU_PORT0
boot_wait3  cmp     ! APU_PORT0
            bne     boot_wait3
            bvs     boot_repeat
            plp
            rts
            end
```

(NCL PG 48)